IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 18, 2018, accepted January 6, 2019, date of publication January 9, 2019, date of current version February 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891770

A Single Attention-Based Combination of CNN and RNN for Relation Classification

XIAOYU GUO[1], HUI ZHANG[1,2], HAIJUN YANG[3], LIANYUAN XU[4], AND ZHIWEN YE[1]

[1]School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China
[3]School of Economics and Management, Beihang University, Beijing 100191, China
[4]School of Energy and Power Engineering, Beihang University, Beijing 100191, China

Corresponding author: Haijun Yang (navy@buaa.edu.cn)

The work of H. Zhang was supported by the National Key R&D Program of China under Grant No. 2017YFB1400200. The work of H. Yang was supported by National Natural Science Foundation of China (Grant No. 71771006).

ABSTRACT As a vital task in natural language processing, relation classification aims to identify relation types between entities from texts. In this paper, we propose a novel Att-RCNN model to extract text features and classify relations by combining recurrent neural network (RNN) and convolutional neural network (CNN). This network structure utilizes RNN to extract higher level contextual representations of words and CNN to obtain sentence features for the relation classification task. In addition to this network structure, both word-level and sentence-level attention mechanisms are employed in Att-RCNN to strengthen critical words and features to promote the model performance. Moreover, we conduct experiments on four distinct datasets: SemEval-2010 task 8, SemEval-2018 task 7 (two subtask datasets), and KBP37 dataset. Compared with the previous public models, Att-RCNN has the overall best performance and achieves the highest F_1 score, especially on the KBP37 dataset.

INDEX TERMS Relation classification, neural network, attention mechanism.

I. INTRODUCTION

As an essential task in NLP, relation classification aims to recognize the semantic relation between two entities in the text based on the predefined class types. Taking the following text as an instance:

> The fifty $[essays]_{e_1}$ collected in this $[volume]_{e_2}$ testify to most of the prominent themes from Professor Quispel's scholarly career.

Where subscripts e_1 and e_2 denote the first and the second entities. The target of relation classification is to identify the relation between ''essay'' and ''volume'', which in this text is ''Member-Collection''.

The past few years have witnessed the validity of deep learning methods, and they are increasingly applied to both unsupervised problems [1], [2] and supervised problems, which include hashing [3], object tracking [4], [5] and classification problems [6]. Especially, deep learning methods looms in computer vision for action proposal [7], [8]. Coincidentally, deep learning methods also attract researchers to relation classification task. Methods dealing with this task can be generally divided into three categories: CNN-based, RNN-based and combined NN-based methods.

On the one hand, a multitude of CNN-based methods have been proposed. Zeng *et al.* [9] extracted sentence level features using CNN and combined these features with handcrafted features to classify relation types. Without dealing with the noisy ''Other'' type in SemEval-2010 task 8 dataset, their model achieved a lower F_1 score than the model proposed by dos Santos *et al.* [10]. They replaced the cross entry loss function with a well-designed pairwise ranking loss function to reduce the impact of noise. As for two subtask datasets from SemEval-2018 task 7, Nooralahzadeh *et al.* [11] employed the shortest dependency path (SDP) information in CNN and obtained a relatively high performance. Nevertheless, by including additional features, such as part-of-speech (POS) features and WordNet-based features, Pratap *et al.* [12] achieved even better performance. On KBP37 dataset, Supervised Ranking CNN [13] with an active learning extension gained the state-of-the-art performance. Although these CNN-based methods are effective because they leverage either a large number of handcrafted features or some prior knowledge, they may also introduce uncontrollable noise from these sources.

On the other hand, approaches based on RNN architecture have also been put forward. The most popular RNN

is LSTM [14], which is capable of learning the long-term dependencies. Typically, Xu *et al.* [15] proposed a model using long short-term memory networks (LSTM) with SDP information added to help discover vital text structures for relation classification. Without treating SDP information as a traditional feature, Cai *et al.* [16] also applied LSTM with SDP encoding into neural networks, which made their model outperform all previous models on SemEval-2010 task 8 dataset. Besides, Zhang and Wang [17] used a simple RNN-based model and achieved a better performance than CNN models on KBP37 dataset. Even though these methods take advantages of automatic feature extracting of RNN, they obtain limited performance because the information extracted by RNN does not contain local features to some extent.

Apart from the methods mentioned above, some of the other works are based on a combination of CNN and RNN to do relation classification task. And some sentence classification works [18] could also be learned from. A comparatively representative approach was proposed by Rotsztejn *et al.* [19], who presented a relation classification system based on an ensemble of CNNs and RNNs. Besides, there are works based on attention [20] which are also popular. Although their method achieves the best performance on three out of four subtasks on SemEval-2018 task 7, methods based on ensemble learning can be more complicated and will occupy much more computing time and resources than single models. Since the scale of a dataset is small, ensemble models not only consume much more time to adjust parameters, but also could be more easily overfitting. Ideally, our target is to construct a deep learning based simple and single model without training multiple models to efficiently and effectively classify relation types.

In this paper, we propose a novel single model Att-RCNN using a combination of CNN and RNN with gated recurrent unit (GRU). Our chief contributions are as follows:

1) Att-RCNN utilizes a combination of both two kinds of NN to capture features by embedding the relation information in texts. In addition, Att-RCNN does not adopt any handcrafted features and achieves better performance in almost all datasets to avoid noise possibly made by the human.
2) We employ multi-level-attention in Att-RCNN. One attention is a word level attention dealing SDP information. The other one is a sentence level attention applying to max pooling procedure. For these two attention mechanisms, we conduct some experiments to reveal that they are complementary to each other and lead to a remarkable improvement.
3) We find a new way to remove noisy text segments in the dataset by introducing SDP information and achieve the state-of-the-art F_1 score of 61.83% on KBP37 dataset. On subtask1.2 dataset of SemEval-2018 task 7, Att-RCNN also outperforms all the single models and achieves a F_1 score of 86.42%, which reaches an increase of 1.5% than the second-ranked model.

The remainder of this paper is structured as follows. Section II presents the materials and methods. Section III provides the datasets and experimental results. Some analysts are summarized and discussed in section IV.

II. MATERIALS AND METHODS

A. MODEL OVERVIEW

To simplify the relation classification task, we try to analyze each text, filter noise and only save key components based on SDP information. After that, we use bidirectional RNN with gated recurrent units (GRU) cells to learn contextual features of each word by using word embeddings as cell input. The output of GRU cells contains information forward and backward. Hence, we combine this kind of information with original word embedding and regard the whole embedding as a representation of a word or text in some way. We then apply a word level attention mechanism and use CNN followed by a sentence level attention to extract the most important and high level features in the text. Eventually, these high level features will be fed into a score computation layer, which includes a class matrix and calculates text scores of every relation class.

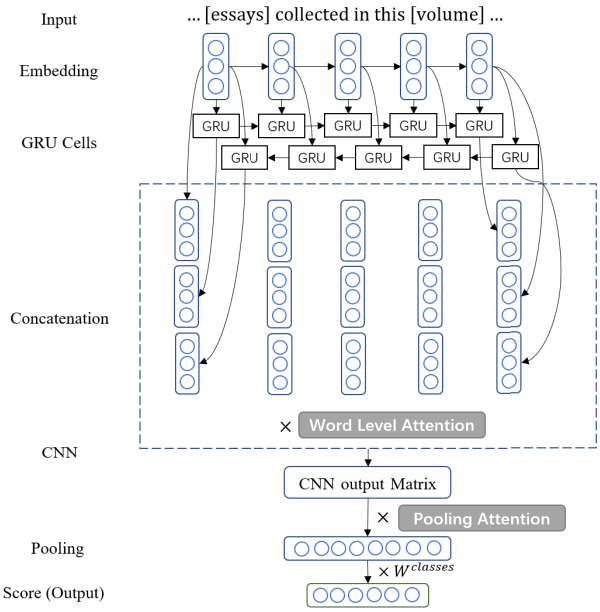


FIGURE 1. Structure of Att-RCNN.

The overall model structure is shown in Fig. 1. And details of Att-RCNN will be covered in following parts of this section.

B. NOISE REMOVING BASED ON SDP INFORMATION

We will introduce our noise removing algorithm in this part. As already stated previously , we need to extract the most important part of texts. So we propose an algorithm to get rid of noise which is harmful in relation classification task. For a text, we firstly analyze semantic dependency tree. Taking the text in section I for example, we only reveal part of the

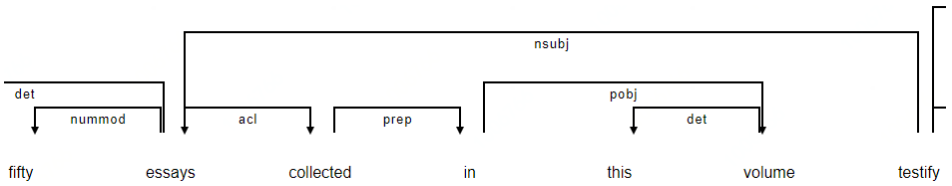


FIGURE 2. An example of dependency tree.

dependency tree of this text in Fig. 2 because the length of example is too long.

As dependency tree showed in this figure, SDP between e_1="essays" and e_2="volume" can be easily got, which is

"*essays* → *collected* → *in* → *volume*".

SDP information contains almost all information of the relation between entities because (1) if entities are arguments of the same predicate, SDP between them will pass through the predicate; (2) if two entities belong to different predicate-argument structures that share a common argument, SDP will pass through this argument [16]. SDP of example text shows that the predicate word "collected" is kept.

However, compared with the original text, SDP information only contains separated and sometimes unrelated words when two entities are far from each other, and will not effectively represent the complete meaning of the text. Under this situation, we will lose essential information that indicates relation information between entities that helps to promote model performance. Because of that, the continuous fragments based on SDP of text are kept with noise removed. And the final input for GRU cells is showed in Fig. 3.

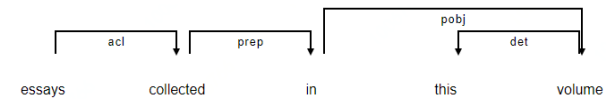


FIGURE 3. An example of final fragment.

By this means, we get a continuous fragment of original text based on SDP information. Although this method of filtering noise and keeping key components turns to have more words left, the average length of text is still shortened by around 50%.

Given a final text fragment $F = \{w_1, w_2, \ldots, w_L\}$, where L is the sentence length. We first get one-hot vector representation v_l of each word and then transform it into a real-valued vector e_l by multiplying an embedding matrix $W_{embeddings} \in \mathbb{R}^{d_w \times V}$, which is:

$$e_l = W_{embeddings} \cdot v_l. \tag{1}$$

where d_w is the dimension of embeddings, and V is the size of our vocabulary. Furthermore, the matrix $W_{embeddings}$ needs to be initialized by pre-trained embeddings and fine-tuned.

C. GRU CELLS

Then, we use bi-GRU [21] to get the context representation of each word. GRU owns less parameters than LSTM, which results in a higher speed in computing convergence. Moreover, based on experiments, GRU gains higher performance than LSTM. We define the left context of word w_l as c_l^{left}, and the right context as c_l^{right}. Both context vectors hold the same dimension just as e_l. By replacing "direct" with "left" or "right", (2) shows the calculation of left or right contextual representation of word w_l.

$$\begin{aligned} r_l^{direct} &= \sigma(W_r^{direct} \cdot e_l + U_r^{direct} \cdot c_{(l-1)}^{direct} + b_r^{direct}) \\ z_l^{direct} &= \sigma(W_z^{direct} \cdot e_l + U_z^{direct} \cdot c_{(l-1)}^{direct} + b_z^{direct}) \\ c_l^{direct} &= (1 - z_l^{direct}) \circ c_{(l-1)}^{direct} \\ &\quad + z_l^{direct} \circ \tanh(W_h^{direct} \cdot e_l \\ &\quad + r_l^{direct} \circ (U_h^{direct} \circ c_{(l-1)}^{direct}) + b_h^{direct}) \end{aligned} \tag{2}$$

where $W_r, U_r, W_z, U_z, W_h, U_h$ are weight matrices, b_r, b_z, b_h are biases and all of them are updated in the process of learning procedure. Operator "$\circ$" donates the Hadamard product. Then we can define the complete word representation of w_l as (3) shows.

$$w_l = [(c_l^{left})^T, (e_l)^T, (c_l^{right})^T]^T \tag{3}$$

where the dimension of c_t^{left} and c_t^{right} is same with the dimension of e_l, which means $w_l \in \mathbb{R}^{3\,d_w}$.

D. WORD LEVEL ATTENTION

Considering the different importance of words in text, we introduce a word level attention to modify the original word contextual representation by multiplying different weights. In detail, words in SDP (exactly in the path) are assigned with a higher weight value and other words are assigned with a lower weight value. And the modified word vectors can be calculated by (4).

$$w_l^{modified} = \begin{cases} \alpha_{high} \times w_l & \text{if } l \in S_{SDP} \\ \alpha_{low} \times w_l & \text{if } l \notin S_{SDP} \end{cases} \tag{4}$$

where S_{SDP} represents words set based on SDP information and α_{high}, α_{low} denote the higher and lower weight values respectively. We employ these two parameters to make our model aware of the differences in words and assign higher weights to words that are of great significance to relations. With a slide window of size k, the complete word representation goes through a CNN for extract the contextual information of text. Specifically, we have:

$$\begin{aligned} R &= [(w_1^{modified})^T, \ldots, (w_T^{modified})^T]^T \\ R^* &= \tanh(W_{CNN} R + B_{CNN}). \end{aligned} \tag{5}$$

where $R \in \mathbb{R}^{3\,d_w \times T}$ is the contextual representation of the full text, and $R^* \in \mathbb{R}^{d_c \times T}$ is the output of CNN. $W_{CNN} \in \mathbb{R}^{d_c \times k(3\,d_w)}$ is a weight matrix with a channel size of d_c.

E. SENTENCE LEVEL ATTENTION

Before applying max pooling to the output of CNN, we introduce a sentence level attention mechanism to strengthen important features in R^* (modify methods by Wang *et al.* [22]).

First of all, we compute a correlation matrix G between each word representation and relation type. In order to achieve that, we introduce auxiliary matrices U and $W^{classes}$. Learned by NNs, U is like a mapping function and converts feature representations of words to relation representations. For each relation $y \in \mathcal{Y}$, we define that $W_y^{classes}$ describes relation class y, which is updated in the training procedure along with other parameters. Combining all these relation class vectors, an embedding matrix $W^{classes}$ whose columns represent different relation classes is what we need. In conclusion, the correlation matrix G is computed by (6).

$$G = R^{*T} U W^{classes} \tag{6}$$

Then a softmax function is applied to compute entries of attention matrix P as

$$P_{i,j} = \frac{\exp(G_{i,j})}{\sum_{j^*=1}^{n} \exp(G_{i,j^*})} \tag{7}$$

where $P_{i,j}$ is the (i,j)-th entry of P, $G_{i,j}$ is the (i,j)-th entry of G. Note that the dimension we apply softmax function is different from Wang *et al.* [22]. This function can strengthen more important features and weaken trivial ones. After that, the attention matrix P is multiplied with the output of convolution layer R^* and goes through a max pooling layer. The i-th entry of output features is calculated as follows in (8).

$$output_i = \max_j (R^* P)_{i,j} \tag{8}$$

where $output_i$ is the i-th entry of $output$ vector and $(R^*P)_{i,j}$ is the (i,j)-th entry of matrix (R^*P).

F. TRAINING OBJECTIVE FUNCTION

We design our loss function following the idea of CR-CNN [10], which could reduce the impact of noisy data. Given output as described above, we compute scores for all relation class in set $\mathcal{Y}$. For $y \in \mathcal{Y}$, the score is computed by (9).

$$s_\theta(F, y) = output^T [W^{classes}]_y \tag{9}$$

where θ denotes all parameters of Att-RCNN. Based on this score function, a pairwise logistic loss function is designed to train Att-RCNN:

$$\begin{aligned}\mathcal{L} = &\log(1 + \exp(\gamma(m^+ - s_\theta(F, y^+)))) \\ &+ \log(1 + \exp(\gamma(m^- + s_\theta(F, y^-)))) + \beta\|\theta\|^2. \end{aligned} \tag{10}$$

Given a text fragment F and its ground truth relation class y^+, $s_\theta(F, y^+)$ is the score for ground truth class, while $s_\theta(F, y^-)$ is the score for a negative class, which is computed by (11). m^+ and m^- are margins which determine thresholds of both correct and incorrect classes. And γ is a factor, which determines the difference between pairwise margins and scores. In addition, $L2$ penalty with the regularization coefficient β is added to prevent overfitting.

$$s_\theta(F, y^-) = \arg\max_{y \in \mathcal{Y}, y \neq y^+} s_\theta(F, y) \tag{11}$$

As it showed in (11), the highest score among the other classes is regarded as the negative score. As the training procedure goes, model increases text scores in ground truth label. In the meanwhile, the score of negative class label will decrease.

III. RESULTS

A. DATASETS

Benefiting from released public benchmark datasets, we can evaluate the performance and robustness of Att-RCNN. We choose four datasets, including SemEval-2010 task 8 dataset, SemEval-2018 task 7 (subtask 1.1 and subtask 1.2) dataset and KBP37 dataset.

1) SEMEVAL-2010 TASK 8

The first dataset we use to evaluate Att-RCNN is SemEval-2010 task 8 dataset. This dataset contains 8000 sentences for training and 2717 sentences for testing. Frequency of relations are listed in Table. 1.

TABLE 1. Frequency of relations in SemEval-2010 task 8 dataset.

Relation	Train dataset	Test dataset
Cause-Effect	1003	328
Component-Whole	941	312
Entity-Destination	845	292
Product-Producer	717	231
Entity-Origin	716	258
Member-Collection	690	233
Message-Topic	634	261
Content-Container	540	192
Instrument-Agency	504	156
Other	1410	454
Total	8000	2717

The first nine relations are directed, and the last one type "Other" is undirected. Thus, there are $9 \times 2 + 1 = 19$ different relations totally. Official macro-averaged F_1-score script is used to evaluate performance of all other models and ours (excluding Other).

2) SEMEVAL-2018 TASK 7

This task published in 2018 contains two subtasks focusing on relation classification. Compared with SemEval-2010 task 8, it deals with semantic relation extraction and classification in scientific papers rather than in common fields. Subtask 1 is released for relation classification, while subtask 2 is used

for relation extraction and classification. Hence, we choose subtask 1 to evaluate Att-RCNN.

The subtask 1 is also built up with two small tasks: one (subtask 1.1) for relation classification on clean data, the other (subtask 1.2) on noisy data. In other words, the entities and relations in subtask 1.1 are annotated by human, while the entities from subtask1.2 are automatically generated by the method proposed by Gábor *et al.* [23]. And the frequencies of relations are listed in Table. 2.

TABLE 2. Frequency of relations in SemEval-2018 task 7.

Relation	Subtask1.1		Subtask1.2	
	Train	Test	Train	Test
USAGE	483	175	470	123
MODEL-FEATURE	326	66	175	75
PART-WHOLE	234	70	196	56
COMPARE	95	21	41	3
RESULT	72	20	123	29
TOPIC	18	3	243	69
Total	1228	355	1248	355

Table. 2 shows that there are total six relations and the whole number of this dataset is much smaller than SemEval-2010 task 8. It is a challenge for deep learning methods including Att-RCNN, because overfitting would be more easily. In order to handle this issue, we adopt a smaller learning rate and a higher penalty to achieve a better performance slightly.

3) THE KBP37 DATASET

This dataset is a revision of MIML-RE annotation dataset [17]. In order to make the dataset more fitful to relation classification task, they made several modifications as follows:

1) First, they add direction to the relation names. That is, each relation is split into two relations which are opposite to each other except for 'no_relation'.
2) Then, the low frequency relations are discarded. Besides, in order to balance the dataset, 80% 'no_relation' are also discarded.
3) After that, the records in dataset are randomly shuffled and divided into three parts: 70% for training, 10% for validation and 20% for testing.

Eventually, eighteen directional relations and one non-directional relation are maintained and the frequencies of them are listed in Table. 3.

Table. 3 reflects that the scale of KBP37 dataset is much larger than the formal two dataset. According to the work of Zhang and Wang [17], this dataset contains more specific entities and relations, which makes it complex and hard to classify the relation types. Because most of the entities in KBP37 are either names of persons, organizations or cities. Besides, there are also some imprecise examples as the relation labels are annotated by human. Hence, these imprecise relation labels may be noise for training and testing and influence the model performance.

TABLE 3. Frequency of relations in KBP37 dataset.

Relation	Train	Valid	Test
per:alternate_names	177	24	46
org:alternate_names	511	63	125
per:origin	266	28	65
org:subsidiaries	832	103	193
per:spouse	258	29	57
org:members	703	82	160
per:title	641	76	137
org:founded	393	53	107
per:employee_of	3472	273	568
org:founded_by	355	34	80
per:countries_of_residence	1660	142	266
org:country_of_headquarters	1006	119	228
per:stateor-provinces_ of_residence	720	66	125
org:stateor-province_ of_headquarters	517	65	126
per:cities_of_residence	663	82	173
org:city_of_headquarters	1267	157	305
per:country_of_birth	355	50	89
org:top_members/employees	576	68	136
no_relation	1545	210	419
Total	15917	1724	3405

B. SETTINGS

We use stochastic gradient descent (SGD) to update model parameters for the first two datasets. To accelerate training procedure, we use Adam [24] on KBP37 dataset. By training skip-gram model [25] on Wikipedia, we get pre-trained word embeddings (WE). Parameters are initialized using a method proposed by Glorot and Bengio [26]. Furthermore, the hyper-parameters are presented in Table. 4.

TABLE 4. Hyper-parameters.

Parameter	2010	2018	KBP37
d_w	300	300	300
d_c	1000	300	1000
k	1	1	1
γ	2	-	2
m^+	2.5	-	2.5
m^-	0.5	-	0.5
β	0.0001	0.0001	0.001
λ	0.01	0.01	0.001

"2010" and "2018" represent SemEval-2010 task 8 dataset and SemEval-2018 task 7 dataset respectively. For different datasets, we keep the dimension of word embeddings d_w, CNN context window size k and loss function factors γ, m^+, m^- constant, but change the size of CNN output channel d_c, normalization coefficient β and learning

TABLE 5. Comparison with other models on SemEval-2010 task 8.

Classifier	Features used	F_1
SVM [27]	POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FramNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner	82.2
CNN+softmax [9]	Word embeddings	69.7
	+ word position embeddings, WordNet	82.7
SDP-LSTM [15]	Word embeddings	82.4
	+ POS, GR, WordNet	83.7
CR-CNN [10]	Word embeddings	82.8
	+ word position embeddings	84.1
depLCNN [28]	Word embeddings, WordNet, word around nominals	83.7
	+ negative sampling from NYT dataset	85.6
BRCNN [16]	Word embeddings	85.4
	+ POS, NER, WordNet	86.3
our model **Att-RCNN**	Word embeddings	**86.6**

rate λ to adapt different sizes. And for SemEval-2018 task 7, we replace the ranking loss function with a simple cross entropy function because there is no "Other" class as noisy relation type according to 10.

There are some hyper-parameters we do not list in Table. 4, which are also important, so we state them separately. First, we apply a dropout to the embedding layer in Fig. 1. The dropout rate is 0.5. Second, for the word level attention, the value of α_h and α_l are assigned according to a proportion of 2:1 on three SemEval-dataset and a proportion of 9:2 on KBP37 dataset. Finally, we introduce learning rate decay λ^* as to reduce initial learning rate λ by (12).

$$\lambda = \lambda^* \times \lambda \tag{12}$$

where λ^* can be regulated differently under distinct circumstances. Since model performance is highly sensitive to these hyper-parameters, we need to adjust them to achieve higher F_1 scores. In order to determine values of these hyper-parameters, we tune these hyper-parameters based on test set for SemEval datasets, since they are small scale. As for KBP37 dataset, which is large scale, we tune hyper-parameters in the validation set.

C. EXPERIMENTS

Since the evaluation of Att-RCNN is on four different datasets mentioned above, we choose models tested on these datasets and bring them into comparison respectively. The following parts of this section will discuss experiments we conduct on these four datasets in detail.

1) RESULTS OF SEMEVAL-2010 TASK 8 DATASET

Experiment results on SemEval-2010 task 8 dataset are listed in Table. 5. The first model [27] in Table. 5 is a feature-based traditional model fed with various handcrafted features, which achieved the highest performance with $F_1 = 82.8\%$ among all traditional methods. Sorted by the public year, the rest of models in Table. 5 are all based on neural networks and all single models.

The experiments show that Att-RCNN outperforms the previous CNN-based and RNN-based models. For example, Att-RCNN achieves nearly 4% higher than CNN+softmax, a CNN-based model and nearly 3% higher than SDP-LSTM, a RNN-based model on F_1 score. In other words, Att-RCNN already exceeds the basic CNN or RNN based models on performance. Furthermore, compared with the well-known and representative CR-CNN model [10], although the objective functions of CR-CNN and Att-RCNN are similar to each other, we improve F_1 score by 2.5% as Att-RCNN takes advantages of both CNN and RNN architecture.

In terms of CNN and RNN combining models, Att-RCNN resembles BRCNN [16] closely. BRCNN makes use of three types of handcrafted features, including POS, NER and WordNet-based features, to introduce informative features and improve model performance. However, Att-RCNN only uses word embeddings to exploit features for relation classification task. Under the same conditions (using only word embeddings), Att-RCNN obtains 1.2% higher than BRCNN on F_1 score. With other features added, we still gain 0.3% higher on F_1 score than the integrated BRCNN model.

2) RESULTS OF SEMEVAL-2018 TASK 7 DATASET

In Table. 6, the F_1 scores we achieve in both subtask 1.1 and subtask 1.2 are compared with other models. Because all of these models are chosen from the CodaLab competition results, we also keep their ranking information in Table. 6.

On subtask 1.1 dataset, we obtain an F_1 score of 64.59%, which is only higher than GU IRLAB. The main reason that Att-RCNN occupies just a mediocre place may lie in the small scale of the dataset. Apart from easily overfitting, replacing the objective function may be to blame.

TABLE 6. Comparison with other models on SemEval-2018 task 7.

Classifier	Subtask 1.1 F_1	rank	Subtask 1.2 F_1	rank
GU IRLAB [29]	60.9	9	78.9	5
MIT-MEDG [30]	72.7	6	80.6	4
SIRIUS-LTG-UiO [11]	76.7	3	83.2	3
Talla (Multi model) [12]	74.2	5	84.8	2
ETH-DS3Lab (Multi model) [19]	81.7	1	90.4	1
our model				
Att-RCNN(Single model)	**64.59**	-	**86.42**	-

On subtask 1.2 dataset, which is mixed with imprecise labeled data, Att-RCNN outperforms almost all these models listed in Table. 6 and all the single models. The experiments reveal that Att-RCNN can even surpass the performance of ensemble learning based model such as Talla, which is a soft-voting ensemble with multi-filter CNN combined. From the point, Att-RCNN obtains over 1.5% higher than Talla in F_1 score. However, the first rank model by ETH-DS3Lab uses multi-combined RNN and CNN, which also introduced a voting strategies to choose the best model. It is not comparable because we aim to construct a simple and single model. Thus, we can conclude that Att-RCNN can effectively deal with noisy data and outperform all the single model on subtask 1.2 dataset.

3) RESULTS OF KBP37 DATASET

In order to comprehensive compare the results achieved on KBP37 dataset, we also evaluate Att-RCNN in view of F_1 score including all relation types. The results are showed in Table. 7.

TABLE 7. Comparison with other models on KBP37 dataset.

Classifier	KBP37 F_1	2010 F_1
CNN+PF [17]	51.3	78.3
CNN+PI [17]	55.1	77.4
RNN+PF [17]	54.3	78.8
RNN+PI [17]	58.8	79.6
BiLSTM-CNN [31]	60.1	81.9
Supervised Ranking CNN [13]	61.26	84.39
our model		
Att-RCNN	**61.83**	**86.6**

Where "PF" represents position features and "PI" stands for position indicators. For KBP37 dataset, we achieve the state-of-the-art result as F_1 score equals 61.83%, which is higher than all previous works. CNN+PF and CNN+PI are simple CNN-based models, while RNN+PF and RNN+PI are simple RNN-based models. Att-RCNN owns at least 3% higher F_1 score than these NN based models.

Besides, BiLSTM-CNN model combines RNN and CNN just like Att-RCNN. But BiLSTM-CNN obtains over 1.7% lower than Att-RCNN model because we reduce the effect of noisy data and choose more valuable features via attention mechanisms. The best result achieved lately by Adilova *et al.* [13] is Supervised Ranking CNN model. By introducing distant supervision and making use of an active learning based extension, Supervised Ranking CNN got rid of noise and obtained good performance on this complex dataset. But it still suffered from noisy labeling of distant supervision, thus was nearly 0.6% lower than Att-RCNN.

To make it more convincing, we also compare the results of SemEval-2010 task 8 with other models. Since our score is 2.21% higher than the Supervised Ranking CNN [13], we can conclude that Att-RCNN could achieve comparable performance to the reference models and single model implement of Att-RCNN is robust.

IV. DISCUSSION AND CONCLUSION

To prove the word level attention and sentence level attention can result in better performance on datasets, we conduct experiments to compare model Att-RCNN and two simplified models, one of which does not contain the word level attention, while the other is only a combination of bi-RNN and CNN. Results are showed in Table. 8.

TABLE 8. Comparison between the main model and simplified models.

Model	2010 F_1	KBP37 F_1
Att-RCNN (origin model)	86.6	61.83
-w/o word level att (model-1)	85.4	55.60
-w/o both attentions (model-2)	85.1	56.08

The experiments show that, F_1-score (model-1) is 85.4% in SemEval-2010 task 8 dataset and 55.60% in KBP37 dataset when we remove the word level attention mechanism. And when we remove all attention mechanisms from origin Att-RCNN model, F_1-score (model-2) continues to decrease on SemEval-2010 task 8 dataset. We observe that two level attention mechanisms promote and interact with each other. Because when we remove both level attentions for model-2, F_1 score is higher than model-1 on KBP37 dataset.

We proposed an Att-RCNN model combining bi-RNN with GRU cells and CNN to improve the performance of the relation classification task. The Att-RCNN model is a single model containing only one layer of RNN with and CNN respectively. Because of that, Att-RCNN could utilize both advantages of RNN and CNN. In order to improve

model performance, we apply two level attention mechanisms to capture more sensible, relevant and valuable features for relation classification task. By evaluating and comparing with other models in the literature, Att-RCNN shows its robustness and overall best performance on four mainstream datasets.

REFERENCES

[1] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.
[2] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.
[3] J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, and R. S. M. Goh, "Transfer hashing: From shallow to deep," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6191–6201, Dec. 2018.
[4] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: A novel multiple-sparse-representation-based tracker," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2022–2037, Apr. 2018.
[5] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa, "Joint sparse representation and robust feature-level fusion for multi-cue visual tracking," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5826–5841, Dec. 2015.
[6] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.
[7] H. Zhu *et al.*, "Yotube: Searching action proposal via recurrent and static regression networks," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2609–2622, Jun. 2018.
[8] H. Zhu, R. Vial, and S. Lu, "Tornado: A spatio-temporal convolutional regression network for video action proposal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5814–5822.
[9] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. 25th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2014, pp. 2335–2344.
[10] C. dos Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proc. 53rd Annu. Meeting Association Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. Assoc. Comput. Linguistics*, vol. 1, 2015, pp. 626–634.
[11] F. Nooralahzadeh, L. Øvrelid, and J. T. Lønning, "Sirius-ltg-uio at semeval-2018 task 7: Convolutional neural networks with shortest dependency paths for semantic relation extraction and classification in scientific papers," in *Proc. 12th Int. Workshop Semantic Eval. Assoc. Comput. Linguistics*, 2018, pp. 805–810.
[12] B. Pratap, D. Shank, O. Ositelu, and B. Galbraith, "Talla at SemEval-2018 task 7: Hybrid loss optimization for relation classification using convolutional neural networks," in *Proc. 12th Int. Workshop Semantic Eval.*, New Orleans, LA, USA, 2018, pp. 863–867.
[13] L. Adilova, S. Giesselbach, and S. Rüping, "Making efficient use of a domain expert's time in relation extraction," *CoRR*, vol. abs/1807.04687, pp. 1–16, Jul. 2018.
[14] J. T. Zhou *et al.*, "SC2Net: Sparse LSTMs for sparse coding," in *Proc. 22nd AAAI Conf. Artif. Intell. (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Edu. Adv. Artif. Intell. (EAAI)*, 2018, pp. 4588–4595.
[15] 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. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1785–1794.
[16] R. Cai, X. Zhang, and H. Wang, "Bidirectional recurrent convolutional neural network for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 756–765.
[17] R. Zhang, F. Meng, Y. Zhou, and B. Liu, "Relation classification viarecurrent neural network with attention and tensor layers," *Big Data Mining Anal.*, vol. 1, no. 3, pp. 234–244, Sep. 2018.
[18] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
[19] J. Rotsztejn, N. Hollenstein, and C. Zhang, "ETH-DS3Lab at SemEval-2018 task 7: Effectively combining recurrent and convolutional neural networks for relation classification and extraction," in *Proc. 12th Int. Workshop Semantic Eval. Assoc. Comput. Linguistics*, 2018, pp. 689–696.
[20] N. Li, H. Zhang, and Y. Chen, "Convolutional neural network with SDP-based attention for relation classification," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 615–618.
[21] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Assoc. Comput. Linguistics*, 2014, pp. 1724–1734.
[22] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention CNNS," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1298–1307.
[23] K. Gábor, H. Zargayouna, D. Buscaldi, I. Tellier, and T. Charnois, "Semantic annotation of the ACL anthology corpus for the automatic analysis of scientific literature," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, N. Calzolari *et al.*, Eds. Paris, France: European Language Resources Association, May 2016, pp. 1–8.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, Dec. 2014.
[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781
[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[27] B. Rink and A. Harabagiu, "UTD: Classifying semantic relations by combining lexical and semantic resources," in *Proc. 5th Int. Workshop Semantic Eval.*, New Orleans, LA, USA, 2010, pp. 256–259.
[28] K. Xu, Y. Feng, S. Huang, and D. Zhao, "Semantic relation classification via convolutional neural networks with simple negative sampling," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2015, pp. 536–540.
[29] S. MacAvaney, L. Soldaini, A. Cohan, and N. Goharian, "GU IRLAB at SemEval-2018 task 7: Tree-LSTMs for scientific relation classification," in *Proc. 12th Int. Workshop Semantic Eval. Assoc. Comput. Linguistics*, 2018, pp. 831–835.
[30] D. Jin, F. Dernoncourt, E. Sergeeva, M. McDermott, and G. Chauhan, "MIT-MEDG at SemEval-2018 task 7: Semantic relation classification via convolution neural network," in *Proc. 12th Int. Workshop Semantic Eval.*, New Orleans, LA, USA, 2018, pp. 798–804.
[31] L. Zhang and F. Xiang, "Relation classification via BiLSTM-CNN," in *Data Mining and Big Data*, Y. Tan, Y. Shi, and Q. Tang, Eds. Cham, Switzerland: Springer, 2018, pp. 373–382.

XIAOYU GUO was born in Hebei, China, in 1993.

He received the B.S. degree in computer science and engineering from Beihang University, China, where he is currently pursuing the master's degree with the State Key Laboratory of Software Development Environment, School of Computer Science.

His main research interests include machine learning, deep learning, and relation classification.

HUI ZHANG was born in Zhejiang, China.

He received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Beihang University, China, where he is currently a Full Professor. His main research interests include Web information search, data mining, and knowledge management.

He is also the Deputy Director of the National Engineering Research Center for Science and Technology Resource Sharing Service, where he is responsible for the integration, management, and sharing of national science and technology project achievements and scientific and technological resources.

HAIJUN YANG was born in Tianjin, China, in 1970.

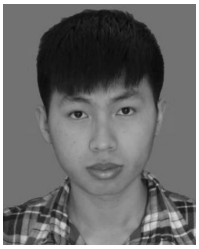
He received the B.S. and M.S. degrees in mathematics and management science from Nankai University, China, and the Ph.D. degree from Tianjin University. He is currently a Full Professor with Beihang University.

He has published more than 30 papers in different journals, such as the *Journal of Evolutionary Economics*, *Entropy*, and the *International Journal of Information Technology & Decision Making*. He is a member of AEA and is also a Fulbright Scholar.

LIANYUAN XU was born in Wuhan, China.

He is currently pursuing the degree in power engineering with Beihang University, Beijing. In 2018, he served as a Wealth Management Intern with GF Securities, Wuhan, and a Quantitative Analyst Intern with Jianghai Securities, Beijing.

He has developed an interest in data mining and is also a Research Assistant with the State Key Laboratory of Software Development Environment, Beihang University.



ZHIWEN YE received the bachelor's degree in engineering, with a major in computer science and technology, from Zhengzhou University, in 2017. He is currently pursuing the master's degree with the State Key Laboratory of Software Development Environment, School of Computer Science, Beihang University.

His main research interests include machine learning, data mining, search engine, and Web information retrieval.