Attention-Based Convolutional Neural Network for Semantic Relation Extraction

Yatian Shen, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, P.R.China
{10110240031,xjhuang}@fudan.edu.cn

Abstract

Nowadays, neural networks play an important role in the task of relation classification. In this paper, we propose a novel attention-based convolutional neural network architecture for this task. Our model makes full use of word embedding, part-of-speech tag embedding and position embedding information. Word level attention mechanism is able to better determine which parts of the sentence are most influential with respect to the two entities of interest. This architecture enables learning some important features from task-specific labeled data, forgoing the need for external knowledge such as explicit dependency structures. Experiments on the SemEval-2010 Task 8 benchmark dataset show that our model achieves better performances than several state-of-the-art neural network models and can achieve a competitive performance just with minimal feature engineering.

1 Introduction

Classifying the relation between two entities in a given context is an important task in natural language processing (NLP). Take the following sentence as an example:

Jewelry and other smaller $\langle e_1 \rangle$ *valuables* $\langle /e_1 \rangle$ *were locked in a* $\langle e_2 \rangle$ *safe* $\langle /e_2 \rangle$ *or a closet with a dead-bolt.*

Here, the marked entities "valuables" and "safe" are of the relation "Content-Container(e1; e2)".

Relation classification plays a key role in various NLP applications, and has become a hot research topic in recent years. Various machine learning based relation classification methods have been proposed for the task, based on either human-designed features (Kambhatla, 2004; Suchanek et al., 2006), or kernels (Kambhatla, 2004; Suchanek et al., 2006). Some researchers also employed the existing known facts to label the text corpora via distant supervision (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012).

All of these approaches are effective because they leverage a large body of linguistic knowledge. However, these methods may suffer from two limitations. First, the extracted features or elaborately designed kernels are often derived from the output of pre-existing NLP systems, which leads to the propagation of the errors in the existing tools and hinders the performance of such systems (Bach and Badaskar, 2007). Second, the methods mentioned above do not scale well during relation extraction, which makes it very hard to engineer effective task-specific features and learn parameters.

Recently, neural network models have been increasingly focused on for their ability to minimize the effort in feature engineering of NLP tasks (Collobert et al., 2011; Zheng et al., 2013; Pei et al., 2014). Moreover, some researchers have also paid attention to feature learning of neural networks in the field of relation extraction. (Socher et al., 2012) introduced a recursive neural network model to learn compositional vector representations for phrases and sentences of arbitrary syntactic types and length. (Zeng et al., 2014; Xu et al., 2015b) utilized convolutional neural networks (CNNs) for relation classification. (Xu et al., 2015c) applied long short term memory (LSTM)-based recurrent neural networks (RNNs) along the shortest dependency path.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2526–2536, Osaka, Japan, December 11-17 2016.

We have noticed that these neural models are all designed as the way that all words are equally important in the sentence, and contribute equally to the representation of the sentence meaning. However, various situations have shown that it is not always the case. For example,

"The $\langle e_1 \rangle$ *women* $\langle /e_1 \rangle$ *that caused the* $\langle e_2 \rangle$ *accident* $\langle /e_2 \rangle$ *was on the cell phone and ran thru the intersection without pausing on the median."*, where the type of relation is "Cause-Effect(e2,e1)".

Obviously, not all words contribute equally to the representation of the semantic relation. In this sentence, "caused" is of particular significance in determining the relation "Cause-Effect", but "phone" is less correlated with the semantic of the relation of "Cause-Effect". So how to identify critical cues which determine the primary semantic information is an important task.

If the relevance of words with respect to the target entities is effectively captured, we can find critical words which determine the semantic information. Hence, we propose to introduce the attention mechanism into a convolution neural network (CNN) to extract the words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector. The key contributions of our approach are as follows:

1. We propose a novel convolution neural network architecture that encodes the text segment to its semantic representation. Compared to existing neural relation extraction models, our model can make full use of the word embedding, part-of-speech tag embedding and position embedding.

2. Our convolution neural network architecture relies on the word level attention mechanism to choose important information for the semantic representation of the relation. This makes it possible to detect more subtle cues despite the heterogeneous structure of the input sentences, enabling it to automatically learn which parts are relevant to the given class.

3. Experiments on the SemEval-2010 Task 8 benchmark dataset show that our model achieves better performance with an F1 score of 85.9% than previous neural network models, and can achieve a competitive performance with an F1 score of 84.3% just with minimal feature engineering.

2 Related Works

A variety of learning paradigms have been applied to relation extraction. As mentioned earlier, supervised methods have shown to perform well in this task. In the supervised paradigm, relation classification is considered as a multi-classification problem, and researchers concentrate on extracting complex features, either feature-based or kernel-based. (Kambhatla, 2004; Suchanek et al., 2006) converted the classification clues (such as sequences and parse trees) into feature vectors. Various kernels, such as the convolution tree kernel (Qian et al., 2008), subsequence kernel (Mooney and Bunescu, 2005) and dependency tree kernel (Bunescu and Mooney, 2005), have been proposed to solve the relation classification problem. (Plank and Moschitti, 2013) introduced semantic information into kernel methods in addition to considering structural information only. However, the reliance on manual annotation, which is expensive to produce and thus limited in quantity has provided the impetus for distant-supervision (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012).

With the recent revival of interest in deep neural networks, many researchers have concentrated on using deep networks to learn features. In NLP, such methods are primarily based on learning a distributed representation for each word, which is also called a word embedding (Turian et al., 2010). (Socher et al., 2012) presented a recursive neural network (RNN) for relation classification to learn vectors in the syntactic tree path connecting two nominals to determine their semantic relationship. (Hashimoto et al., 2013) also employed a neural relation extraction model allowing for the explicit weighting of important phrases for the target task. (Zeng et al., 2014) exploited a convolutional deep neural network to extract lexical and sentence level features. These two levels of features were concatenated to form the final feature vector. (Ebrahimi and Dou, 2015) rebuilt an RNN on the dependency path between two marked entities. (Xu et al., 2015b) used the convolutional network and proposed a ranking loss function with data cleaning. (Xu et al., 2015c) leveraged heterogeneous information along the shortest dependency path between two entities. (Xu et al., 2016) proposed a data augmentation method by leveraging the directionality of relations.

Another line of research is the attention mechanism for deep learning. (Bahdanau et al., 2014) pro-

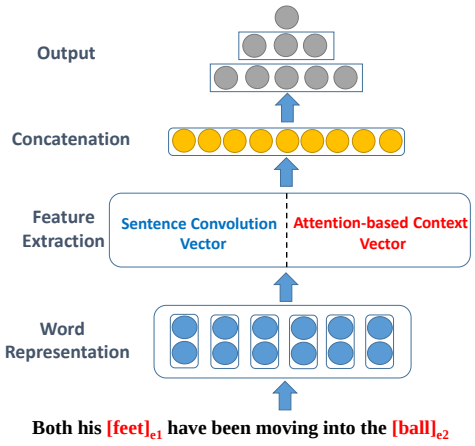


Figure 1: Architecture of the attention-based convolution neural network.

posed the attention mechanism in machine translation task, which is also the first use of it in natural language processing. This attention mechanism is used to select the reference words in the original language for words in the foreign language before translation. (Xu et al., 2015a) used the attention mechanism in image caption generation to select the relevant image regions when generating words in the captions. Further uses of the attention mechanism included paraphrase identification (Yin et al., 2015), document classification (Yang et al., 2016), parsing (Vinyals et al., 2015), natural language question answering (Sukhbaatar et al., 2015; Kumar et al., 2015; Hermann et al., 2015) and image question answering (Lin et al., 2015). (Wang et al., 2016) introduced attention mechanism into relation classification which relied on two levels of attention for pattern extraction. In this paper, we will explore the word level attention mechanism in order to discover better patterns in heterogeneous contexts for the relation classification task.

3 Methodology

Given a set of sentences $x_1, x_2, ...x_n$ and two corresponding entities, our model measures the probability of each relation r. The architecture of our proposed method is shown in Figure 1. Here, feature extraction is the main component, which is composed of sentence convolution and attention-based context selection. After feature extraction, two kinds of vectors – the sentence convolution vector and the attention-based context vector, are generated for semantic relation classification.

- **Sentence Convolution**: Given a sentence and two target entities, a convolutional neutral network (CNN) is used to construct a distributed representation of the sentence.
- **Attention-based Context Selection**: We use word-level attention to select relevant words with respect to the target entities.

3.1 Sentence Convolution

3.1.1 Input of Model

Word Embeddings. Figure 2 shows the architecture of our convolution neural network. In the word representation layer, each input word token is transformed into a vector by looking up word embeddings. (Collobert et al., 2011) reported that word embeddings learned from significant amounts of unlabeled data are far more satisfactory than the randomly initialized embeddings. Although it usually takes a long time to train the word embeddings, there are many freely available trained word embeddings. A comparison of the available word embeddings is beyond the scope of this paper. Our experiments directly utilize the embeddings trained by the CBOW model on 100 billion words of Google News (Mikolov et al., 2013).

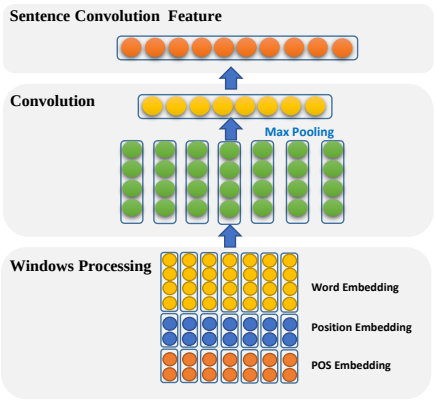


Figure 2: Architecture of convolution neural network.

Position Embeddings. In the task of relation extraction, the words close to the target entities are usually more informative in determining the relation between entities. Similar to (Zeng et al., 2014), we use position embeddings specified by entity pairs. It can help the CNN to keep track of how close each word is to the head or the tail entity, which is defined as the combination of the relative distances from the current word to the head or the tail entity. For example,

"The $\langle e_1 \rangle$ *game* $\langle /e_1 \rangle$ *was sealed in the original* $\langle e_2 \rangle$ *packing* $\langle /e_2 \rangle$ *unopened and untouched."*

In this sentence, the relative distance from the word "sealed" to the head entity "game" is 2 and the tail entity "packing" is -4. According to the above rule, we can obtain the relative distance from every word in the above sentence to each entity. We first create two relative distance files of entity e_1 and entity e_2. Then, we use the CBOW model to pretrain position embeddings on two relative distance files respectively (Mikolov et al., 2013). The dimension of position embedding is set 5.

Part-of-speech tag Embeddings. Our word embeddings are obtained from the Google News corpus, which is slightly different to the relation classification corpus. We deal with this problem by allying each input word with its POS tag to improve the robustness. In our experiment, we only take into use a coarse-grained POS category, containing 15 different tags. We use the Stanford CoreNLP Toolkit to obtain the part-of-speech tagging (Manning et al., 2014) . Then we pretrain the embeddings by the CBOW model on the taggings, and the dimension of part-of-speech tag embedding is set 10.

Finally, we concatenate the word embedding, position embedding, and part-of-speech tag embedding of each word and denote it as a vector of sequence $w = [WF, pF, POSF]$.

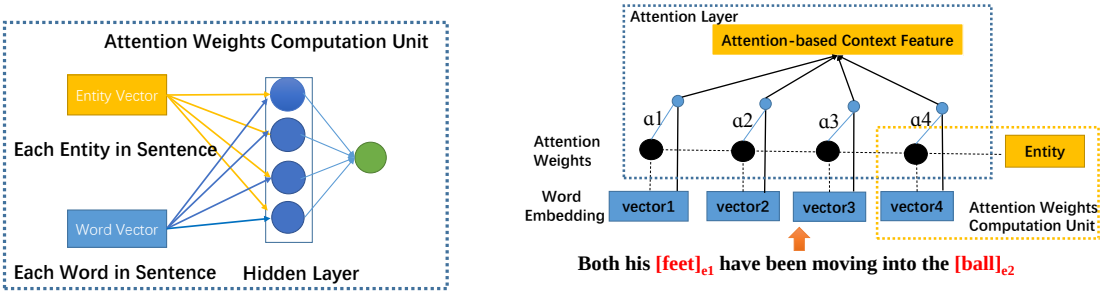
3.1.2 Convolution, Max-pooling and Non-linear Layers

In relation extraction, one of the main challenges is that, the length of the sentences is variable and important information can appear anywhere. Hence, we should merge all local features and perform relation prediction globally. Here, we use a convolutional layer to merge all these features. The convolutional layer first extracts local features with a sliding window of length l over the sentence. We assume that the length of the sliding window l is 3. Then, it combines all local features via a max-pooling operation to obtain a fixed-sized vector for the input sentence. Since the window may be outside of the sentence boundaries when it slides near the boundary, we set special padding tokens for the sentence. It means that we regard all out-of-range input vectors w_i ($i < 1$ or $i > m$) as zero vector.

Let $x_i \in R^k$ be the k-dimensional input vector corresponding to the ith word in the sentence. A sentence of length n (padded where necessary) is represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus x_3 \oplus ... \oplus x_n \tag{1}$$

where $\oplus$ is the concatenation operator. Let $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, ..., x_{i+j}$. A convolution operation involves a filter $w \in R^{hk}$, which is applied to a window of h words to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by



(a) Attention weights computation unit.

(b) Attention layer network.

Figure 3: Architecture of attention layer network.

$$c_i = f(w \cdot x_{i:i+h-1}) \tag{2}$$

Here f is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ to produce a feature map:

$$\mathbf{c} = [c_1, c_2, ..., c_{n-h+1}] \tag{3}$$

with $c \in R^{n-h+1}$. We then apply a max-overtime pooling operation over the feature map and take the maximum value $\hat{c} = max\{\boldsymbol{c}\}$ as the feature. The idea is to capture the most important feature – one with the highest value – for each feature map. This pooling scheme naturally deals with variable sentence lengths.

3.2 Attention-based Context Selection

Our attention model is applied to a rather different kind of scenario, which consist of heterogeneous objects, namely a sentence and two entities. So we seek to give our model the capability to determine which parts of the sentence are most influential with respect to the two entities of interest. For instance,

"That coupled with the $\langle e_1 \rangle$ *death* $\langle /e_1 \rangle$ *and destruction caused by the* $\langle e_2 \rangle$ *storm* $\langle /e_2 \rangle$ *was a very traumatic experience for these residents."*.

Here, the type of relation is "Cause-Effect(e2,e1)".

In this sentence, the non-entity word "caused" is of particular significance in determining the relation "Cause-Effect". Fortunately, we can exploit the fact that there is a salient connection between "caused" and "death". We introduce a word attention mechanism to quantitatively model such contextual relevance of words with respect to the target entities.

In order to calculate the weight of each word in the sentence, we need to feed each word in the sentence and each entity to a multilayer perceptron (MLP). The network structure of the attention weight computation is shown in Figure 3 (a).

Assume that each sentence contains T words. w_{it} with $t \in [1, T]$ represents the words in the ith sentence. e_{ij} with $j \in [1, 2]$ represents the jth entity in the ith sentence. We concatenate the representation of entity e_{ij} and the representation of word w_{it} to get a new representation of word t, i.e., $h_{it}^j = [w_{it}, e_{ij}]$. u_{it}^j quantifies the degree of relevance of the tth word with respect to the jth entity in the ith sentence. This relevance scoring function is computed by the MLP network between the respective embeddings of the word w_{it} and the entity e_{ij}. We named the degree of relevance as the word attention weight, namely, u_{it}^j. The calculation procedure of u_{it}^j is as follows:

$$h_{it}^j = [w_{it}, e_{ij}] \tag{4}$$

$$u_{it}^j = W_a[\tanh(W_{we}h_{it}^j + b_{we})] + b_a \tag{5}$$

The output of the attention MLP network is u_{it}^{j}. Now we can get a normalized importance weight α_{it}^{j} through a softmax function.

$$\alpha_{it}^{j} = \frac{\exp(u_{it}^{j})}{\sum_{t} \exp(u_{it}^{j})} \tag{6}$$

The architecture of our proposed attention layer is shown in Figure 3 (b). After that, we compute the sentence context vector s_{ij} about entity j as a weighted sum of the word in the sentence i based on the weights as follows:

$$s_{ij} = \sum_{t} \alpha_{it}^{j} w_{it} \tag{7}$$

The context vector s_{ij} can be seen as a high level representation of a fixed query "what is the informative word" over the words. The weight of attention MLP network is randomly initialized and jointly learned during the training process.

3.3 MLP Layer

At last, we can obtain the output of three networks, which includes the result of convolution network, and the sentence context vectors of the two entities. We then concatenate all three output vectors into a fixed-length feature vector.

The fixed length feature vector is fed to a multi-layer perceptron (MLP), which is shown in Figure 1. More specifically, first, the vector obtained is fed into a full connection hidden layer to get a more abstractive representation, and then, this abstractive representation is connected to the output layer. For the task of classification, the outputs are the probabilities of different classes, which is computed by a softmax function after the fully-connected layer. We name the entire architecture of our model **Attention-CNN**.

3.4 Model Training

The relation classification model proposed here using attention-based convolutional neural network could be stated as a parameter vector θ. To obtain the conditional probability $p(i|x, \theta)$, we apply a softmax operation over all relation types:

$$p(i|x, \theta) = \frac{e^{o_i}}{\sum_{k=1}^{n} e^{o_k}} \tag{8}$$

Given all the T training examples $(x^{(i)}; y^{(i)})$, we can then write down the log likelihood of the parameters as follows:

$$\mathcal{J}(\theta) = \sum_{i=1}^{T} \log p(y^{i}|x^{i}, \theta) \tag{9}$$

To compute the network parameter of θ, we maximize the log likelihood $\mathcal{J}$ using stochastic gradient descent (SGD). θ are randomly initialized. We implement the back-propagation algorithm and apply the following update rule:

$$\theta \leftarrow \theta + \lambda \frac{\partial \log p(y|x, \theta)}{\partial \theta} \tag{10}$$

Minibatch size	32
Word embedding size	300
Word Position Embedding size	5
Part-of-speech tag Embeddings	10
Word Window size	3
Convolution size	100
Learning rate	0.02

Table 1: Hyperparameters of our model

4 Experiments

4.1 Dataset and Evaluation Metrics

We evaluated our model on the SemEval-2010 Task 8 dataset, which is an established benchmark for relation classification (Hendrickx et al., 2009). The dataset contains 8000 sentences for training, and 2717 for testing. We split 1000 samples out of the training set for validation.

The dataset distinguishes 10 relations, and the former 9 relations are directed, whereas the "Other" class is undirected. In our experiments, We do not distinguish the direction of the relationship. To compare our results with those obtained in previous studies, we adopt the macro-averaged F1-score in our following experiments.

4.2 Parameter Settings

In this section, we experimentally study the effects of different kinds of parameters in our proposed method: Word embedding size, Word Position Embedding size, Word Window size, Convolution size, Learning rate, and Minibatch size. For the initialization of the word embeddings used in our model, we use the publicly available word2vec vectors that were trained on 100 billion words from Google News. Words not present in the set of pre-trained words are initialized randomly. The other parameters are initialized by randomly sampling from the uniform distribution in [-0.1,0.1].

Model	Feature Sets	F_1
SVM	POS, stemming, syntactic pattern, WordNet	78.8
RNN	-	74.8
	+POS, NER, WordNet	77.6
MVRNN	-	82.4
	+POS, NER, WordNet	82.4
CNN	-	78.9
(Zeng et al., 2014)	+WordNet, words around nominals	82.7
FCM	depedency parsing, NER	83.0
CR-CNN	+WordNet, words around nominals	83.7
SDP-LSTM	POS, WordNet, grammar relation	83.7
	-	**84.3**
Attention-CNN	+WordNet, words around nominals	**85.9**

Table 2: Comparison of the proposed method with existing methods in the SemEval-2010 Task 8 dataset.

For other hyperparameters of our proposed model, we take those hyperparameters that achieved the best performance on the development set. The final hyper-parameters are shown in Table 1.

4.3 Results of Comparison Experiments

To evaluate the performance of our automatically learned features, we select six approaches as competitors to be compared with our method.

Table 2 summarizes the performances of our model, SVM (Hendrickx et al., 2009), RNN, MV-RNN (Socher et al., 2012), CNN (Zeng et al., 2014), FCM (Gormley et al., 2015), CR-CNN (Xu et al., 2015b), and SDP-LSTM (Xu et al., 2015c). All of the above models adopt word embedding as representation except SVM. For fair comparison among the different model, we also add two types of lexical features, WordNet hypernyms and words around nominals, as part of the fixed length feature vector to the MLP layer.

We can observe in Table 2 that, Attention-CNN, without extra lexical features such as WordNet and words around nominals, still outperforms previously reported best systems of CR-CNN and SDP-LSTM with F1 of 83.7%, though both of which have taken extra lexical features into account. It shows that our method can learn a robust and effective relation representation. When added with the same lexical features, our Attention-CNN model obtains the result of 85.9%, significantly better than CR-CNN and

SDP-LSTM. In general, richer feature sets lead to better performance. Such neural models as RNN, MVRNN, CR-CNN and SDP-LSTM can automatically learn valuable features, and all of these models heavily depend on the result of the syntactic parsing. However, the error of syntactic parsing will inevitably inhibit the ability of these methods to learn high quality features.

Similarly, Attention-CNN, CNN, and CR-CNN all apply convolution neural network to the extraction of sentence features, but we can see from Table 2 that Attention-CNN yield a better performance of 84.3%, compared with CNN and CR-CNN. One of the reason is that the input of the three models are different. Our model uses word embeddings, position embeddings, part-of-speech embeddings as input. CNN also leverages position embeddings and lexical features. CR-CNN makes use of heterogeneous information along the shortest dependency path between two entities. Our experiments verify that the part-of-speech embeddings used by us contain rich semantic information. On the other hand, our proposed Attention-CNN model can still yield higher F1 without prior NLP knowledge. The reason should be due to that word level attention mechanism is able to better choose which parts of the sentence are more discriminative with respect to the two entities of interest.

Feature Sets	F_1
WF	74.5
+pF	80.7
+POSF	82.6
+WA	84.3
+WA+(Lexical Feature)	85.9

Table 3: Score obtained for various sets of features on the test set. The bottom portion of the table shows the best combination of all the features.

4.4 Effect of Different Feature Component

Our network model primarily contains four sets of features, "Word Embeddings (WF)","Position Embeddings (pF)","Part-of-speech tag Embeddings (POSF)", and "Word Attention (WA)". We performed ablation tests on the four sets of features in Table 3 to determine which type of features contributed the most. From the results we can observe that our learned position embedding features are effective for relation classification. The F1-score is improved remarkably when position embedding features are added. POS tagging embeddings are comparatively more informative, which can boost the F1 by 1.9%. The system achieves approximately 2.3% improvements when adding Word Attention. When all features are combined, we achieve the best result of 85.9%.

4.5 Visualization of Attention

In order to validate whether our model is able to select informative words in a sentence or not, we visualize the word attention layers in Figure 4 for several data from test sets.

Every line in Figure 4 shows a sentence. The size of a word denotes the importance of it. We normalize the word weight to make sure that only important words are emphasized. Given the following sentence as an example,

"The burst has been caused by water hammer pressure."

we can find that the word "caused" was assigned the highest attention score, while words such as "burst" and "pressure" also are important. This makes sense in light of the ground-truth labeling as a "Cause-Effect" relationship. Additionally, we observe that words like "The", "has" and "by" have low attention scores. These are indeed rather irrelevant with respect to the "Component-Whole" relationship.

5 Conclusion

In this paper, we propose an attention-based convolutional neural network architecture for semantic relation extraction. Here, the convolutional neural network architecture is used to extract the features

Relation	Representation of Word Attention Weight
Instrument-Agency	The author of a keygen uses a disassembler to look at the raw assembly code
Message-Topic	The Pulitzer Committee issues an official citation explaining the reasons for the award
Cause-Effect	The burst has been caused by water hammer pressure
Instrument-Agency	Even commercial networks have moved into high-definition broadcast
Component-Whole	The girl showed a photo of apple tree blossom on a fruit tree in the Central Valley
Member-Collection	They tried an assault of their own an hour later, with two columns of sixteen tanks backed by a battalion of Panzer grenadiers

Figure 4: Visualization of Attention.

of the sentence. Our model can make full use of word embedding, part-of-speech tag embedding and position embedding information. Meanwhile, word level attention mechanism is able to better determine which parts of the sentence are most influential with respect to the two entities of interest. Experiments on the SemEval-2010 Task 8 benchmark dataset show that our model achieves better performances than several state-of-the-art systems.

In the future, we will focus on exploring better neural network structure about feature extraction in relation extraction. Meanwhile, because end-to-end relation extraction is also an important problem, we will seek better methods for completing entity and relation extraction jointly.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This work was partially funded by National Natural Science Foundation of China (No. 61532011 and 61472088), the National High Technology Research and Development Program of China (No. 2015AA015408).

References

Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Literature review for Language and Statistics II*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Javid Ebrahimi and Dejing Dou. 2015. Chain based rnn for relation classification. In *Proceedings of the Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL: Association for Computational Linguistics*, pages 1244–1249.

Matthew R Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. *arXiv preprint arXiv:1505.02419*.

Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Simple customization of recursive neural networks for semantic relation classification. In *EMNLP*, pages 1372–1376.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Raymond J Mooney and Razvan C Bunescu. 2005. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178.

Wenzhe Pei, Tao Ge, and Chang Baobao. 2014. Maxmargin tensor neural network for chinese word segmentation. In *Proceedings of ACL*.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *ACL (1)*, pages 1498–1507.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 697–704. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 712–717. ACM.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 721–729. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015a. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2(3):5.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015b. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015c. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (to appear)*.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. *arXiv preprint arXiv:1601.03651*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *EMNLP*, pages 647–657.