Effective Deep Memory Networks for Distant Supervised Relation Extraction

Xiaocheng Feng, Jiang Guo, Bing Qin, Ting Liu, Yongjie Liu
SCIR, Harbin Institute of Technology, Harbin, China
{xcfeng, jguo, qinb, tliu, yongjieliu}@ir.hit.edu.cn

Abstract

Distant supervised relation extraction (RE) has been an effective way of finding novel relational facts from text without labeled training data. Typically it can be formalized as a multi-instance multi-label problem. In this paper, we introduce a novel neural approach for distant supervised RE with special focus on attention mechanisms. Unlike the feature-based logistic regression model and compositional neural models such as CNN, our approach includes two major attention-based memory components, which are capable of explicitly capturing the importance of each context word for modeling the representation of the entity pair, as well as the intrinsic dependencies between relations. Such importance degree and dependency relationship are calculated with multiple computational layers, each of which is a neural attention model over an external memory. Experiment on real-world datasets shows that our approach performs significantly and consistently better than various baselines.

1 Introduction

Relation extraction (RE) aims at extracting semantic relations between entities. Formally, given a sentence s with the annotated head entity e_h and tail entity e_t, the goal of RE is to predict the relations between e_h and e_t. RE is a fundamental task in Natural Language Processing (NLP), and a crucial component for building structured Knowledge Base (KB) from free texts. Previous methods including feature-based approaches [Zhang *et al.*, 2006; Li *et al.*, 2012] and neural-based approaches [Nguyen and Grishman, 2015b] can extract high-quality relational facts based on human annotations. However, the heavy cost of annotation typically limits the existing labeled data of RE in both scale and domains.

A promising RE paradigm that addresses this challenge is distant supervision, which can automatically generate training data by aligning a database of relational facts with text [Mintz *et al.*, 2009]. Figure 1 shows a simple example for a RE domain with two labels. Distant supervision will regard all sentences (S1, S2, S3) that contain these two entities as active instances. Therefore, it can be formalized as a multi-instance multi-label classification problem.

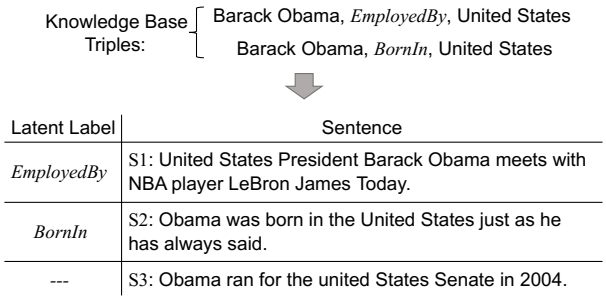

Figure 1: Training sentences generated through distant supervision for a knowledge base containing two facts.

To facilitate the modeling of texts, neural networks have been widely explored in distant supervised RE and achieved state-of-the-art results [Zeng *et al.*, 2015; Lin *et al.*, 2016; Jiang *et al.*, 2016]. Various neural networks such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have been adopted [Zeng *et al.*, 2014; Xu *et al.*, 2015; Vu *et al.*, 2016] to learn the representation of each instance (sentences), which is then used as the representation of the corresponding entity pair for relation classification.

In this paper, we investigate effective neural attention mechanisms for distant supervised RE. We base our approach on the following two observations:

- Not all context words contribute equally to the inference of relation for an entity pair. For example, in S2, "born" is an important clue for the entity pair (Barack Obama, United States) while "said" is much less important.
- There exists dependencies (e.g., entailment, conflict) between different relations, which is a crucial cue to infer some instances with implicit relation expression. For instance, if triple (A, capital, B) holds, another triple (A, contains, B) will hold as well.

Therefore, a desirable solution should not only have the capability of explicitly capturing the importance of different context words but also automatically learning the dependencies between relations.

In pursuit of these goals, we propose a neural model that includes two attention-based memory networks inspired by the recent success of computational models with explicit memory [Sukhbaatar *et al.*, 2015; Tang *et al.*, 2016;

Lin *et al.*, 2016]. The first one is a a word-level memory network, which learns the importance/weight of each context word with regard to the specific entity pair. Then, the entity pair representation will be computed as the semantic composition of the soft-attended context words. Afterwards, we present a two-layer relation-level attention-based memory network. The first layer is to capture the weight of each active instance, which addresses the wrong labeling problem [Lin *et al.*, 2016]. The second layer is to learn the dependencies between relations. The output of the second layer will be used as the representation of the bag-of-instances for multi-label relation classification. As every component is differentiable, the entire model can be efficiently trained end-to-end with gradient descent. Experimental results show that our model achieve significant and consistent improvements in relation extraction as compared with the state-of-the-art methods.

Our contributions can be summarized as follows:

- We present a novel neural architecture with two memory networks, which is capable of modeling the semantic relatedness of the entity pair with its contexts words as well as the dependencies between relations.
- Experiments on real-world datasets show that our approach significantly and consistently outperforms all baselines.

2 Background

2.1 Memory Network

Memory network is a general machine learning framework introduced by [Weston *et al.*, 2014]. Its central idea is inference with a long-term memory component, which could be read, written to, and jointly learned with the goal of using it for prediction. Formally, a memory network consists of a memory m and four components *I*, *G*, *O* and *R*, where m is an array of objects such as an array of vectors. Let us take question answering as an example to explain the work flow of memory network. Given a list of sentences and a question, the task aims to find evidences from these sentences and generate an answer, e.g. a word. During inference, *I* component reads one sentence s_i at a time and encodes it into a vector representation. Then *G* component updates a piece of memory m_i based on current sentence representation. After all sentences are processed, we get a memory matrix m which stores the semantics of these sentences, each row representing a sentence. Given a question q, memory network encodes it into vector representation e_q, and then *O* component uses e_q to select question related evidences from memory m and generates an output vector o. Finally, R component takes o as the input and outputs the final response. [Sukhbaatar *et al.*, 2015; Rush *et al.*, 2015; Tang *et al.*, 2016] demonstrate that multiple hops could uncover more abstractive evidences than single hop, and could yield improved results on question answering , summarization and sentiment classification.

2.2 Convolutional Neural Network

In this section, we briefly introduce how to model entity pair semantic representation with CNN, as widely used in previous studies [Nguyen and Grishman, 2015a; Zeng *et al.*, 2014]. A CNN with three convolutional filters is illustrated in Figure 2.

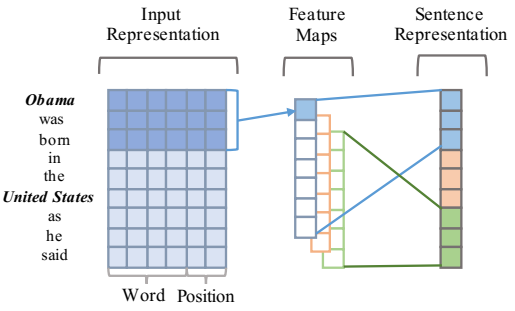

Figure 2: Sentence-level feature extraction using CNN.

Denote a sentence consisting of n words as $\{w_1, w_2, ...w_i, ...w_n\}$, and each word w_i is mapped to its embedding $e_i \in \mathbb{R}^d$. An additional position feature (PF) is used to indicate the relative distance of the current word to the two entities. A convolutional filter is a list of linear layers with shared parameters. Let l_{cf} be the width of a convolutional filter, and W_{cf} , b_{cf} be the shared parameters of linear layers in the filter. The input of a linear layer is the concatenation of word embeddings in a fixed-length window size l_{cf}. The output of a linear layer is calculated as $O_{cf} = W_{cf} \cdot l_{cf} + b_{cf}$, where $W_{cf} \in \mathbb{R}^{d \times l_{cf}}$ and $b_{cf} \in \mathbb{R}^{len}$. len is the output length of the linear layer. The output of each convolutional filter is fed to a MaxPooling layer, resulting in an output vector with fixed length.

3 Deep Memory Networks for Distant Supervised Relation Extraction

This section describes our deep memory network approach for distant supervised RE. We first give the task definition and notations, following with an overview of our proposed neural architecture. Then, we provide detailed formalizations of our model in distant supervised RE, with special emphasize on the two major memory components.

3.1 Task Definition and Notation

Given a set of sentences $S = \{s_1, ..., s_i, ..., s_n\}$ consisting of n sentences and an entity pair $\{e_h, e_t\}$ [1] occurring in all sentences, distant supervised RE aims at predicting the relation of sentence set S towards the entity pair $\{e_h, e_t\}$. For example, in figure 1, entity pair ("Obama", "United States") has two labels, "*EmployedBy*" and "*BornIn*".

When dealing with a text corpus, we map each word into a low dimensional, continuous and real-valued vector, which is also known as word embeddings [Mikolov *et al.*, 2013; Pennington *et al.*, 2014]. All the word vectors are stacked in a word embedding matrix $L \in \mathbb{R}^{d_w \times |V|}$, where d_w is the

[1] In practice, an entity might be a multi-word expression such as "United States". For simplicity, we still consider entity as a single word in this definition. And their entity representation is an average of its constituting word vectors [Sun *et al.*, 2015].

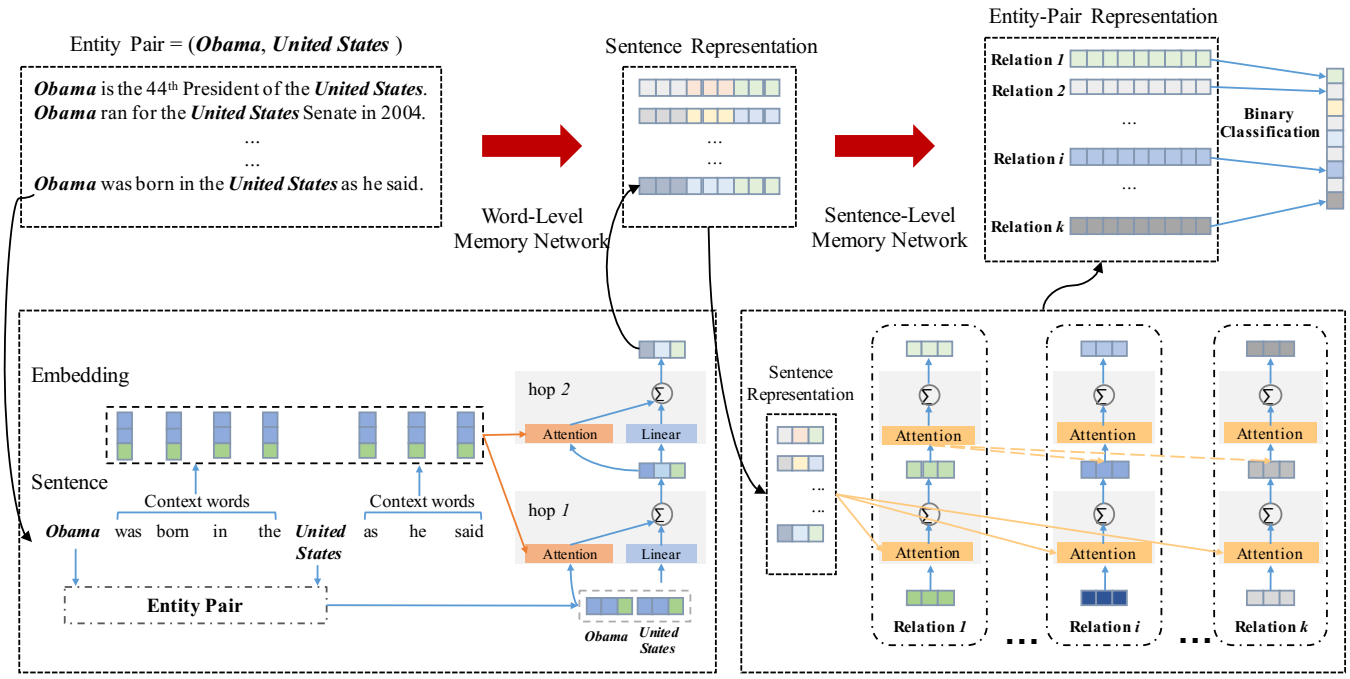

Figure 3: Illustration of our deep memory networks with two computational layers (hops) for distant supervised RE.

dimension of word vector and $|V|$ is vocabulary size. The embedding of the i^{th} word is notated as $e_i \in \mathbb{R}^{d_w \times 1}$, which is a column in the embedding matrix L. In addition, to specify the position of each entity pair, we also use position embeddings for all words in the sentence. Each content word has two relative positions to head entity e_h and tail entity e_t, and they are mapped to two different d_p dimensional vectors separately. For example in S1 from Figure 1, the relative distance from the word "President" to head entity "Obama" is -2 and tail entity 'United States' is 1. Finally, we concatenate the word embeddings and position embeddings of all words and denote it as a vector sequence $s = \{e_1, e_2, ..., e_i, ..., e_n\}$, where $e_i \in \mathbb{R}^{d \times 1} (d = d_w + 2 \times d_p)$.

3.2 An Overview of the Approach

In this section, we present an overview of our model for distant supervised RE, as illustrated in Figure 3. Specifically, our model includes two major components, both of which are built with deep memory networks.

Word-Level Memory Network Given a sentence $s = \{w_1, w_2, ..., w_i, ..., w_l\}$ and an entity pair, a word-level memory network is used to model a distributed representation v_s of this sentence towards the entity pair. To simplify the interpretation, we consider entity pair as two single words w_h and w_t. Context word vectors $\{e_1, ..., e_{e_h-1}, e_{e_h+1}, ..., e_{e_t-1}, e_{e_t+1}, ..., e_l\}$ are stacked and regarded as the external memory $l \in \mathbb{R}^{d \times (l-2)}$, where l is the sentence length.

An illustration of this network is given in bottom-left dashed box of Figure 2. In the first computational layer (hop 1), we use the entity vector as input to adaptively select important evidences from memory m through an attention layer. The output of the attention layer and the linear transformation of entity pair vector are summed and fed to the next layer (hop 2). In a similar way, we stack multiple hops and run these steps multiple times, so that more abstractive evidences could be selected from the external memory m. The output vector in the last hop is concatenated with the output of CNN model (Section 2.2) and the resulting vector is considered as the representation of sentence with regard to the entity pair.[2]

Relation-Level Memory Network Suppose there is a set $S = \{s_1, ..., s_i, ..., s_n\}$ contains n sentences for entity pair (bag-of-instances) and each sentence s_i has a distributed representation x_i. We further propose a relation-level memory network to generate a representation of the sentence set towards the entity pair. Specifically, for different relations, each entity pair has a different representation.

As shown in the bottom-right dashed box of Figure 2, we build a two-layer memory network for each relation. In the first layer, we follow [Lin *et al.*, 2016] and use instance-level attention to select the sentences which really express the corresponding relation. For relation r_i-related memory network , we randomly generate a vector v_{r_i} which indicates the representation of relation r_i and use it to calculate the relevance weight between relation r_i and each sentence through a attention layer. In the second layer, we develop a relation-level attention model to learn the dependencies between relations. For instance, if a person is a founder of company (A, founder, B), we will know that A has high probability to be a major shareholder of B (A, major_shareholders, B). The input of this

[2]It is helpful to note that the parameters of attention and linear layers are shared in different hops.

layer is the output of the first attention layer and this input will be used to calculate the dependency weights between relation r_i and other relation $\{r_1, ..., r_{i-1}, r_{i+1}, ..., r_k\}$. Finally, the resulting representation is used for multi-label relation classification.

3.3 Word Attention

Here we introduce the word-attention model. The basic idea of attention mechanism is that it assigns a weight (importance) to each lower position when computing an upper level representation (Bahdanauet al., 2015). In this work, we use a word-attention model to compute the representation of a sentence with regard to an entity pair. The intuition is that context words do not contribute equally to the semantic meaning of a sentence. Furthermore, the importance of a word should be different if we focus on different entity pairs. Let us take the sentences "US President Barack Obama meets NBA player LeBron James" as an example. The context word "President" is more important than "player" for entity pair ("US", "Barack Obama"). On the contrary, "player" is more important than "President" for entity pair ("NBA", "LeBron James").

Taking an external memory $m \in \mathbb{R}^{d \times w}$ and an entity pair vector $V_{EP} \in \mathbb{R}^{2d \times 1}$ (concatenating two entities) as input, the attention model outputs a continuous vector $x \in \mathbb{R}^{d \times 1}$, which is a weighted sum of each piece of memory in m:

$$x = \sum_{i=1}^{w} \alpha_i m_i \tag{1}$$

where w is the memory size, $\alpha_i \in [0, 1]$ is the weight of m_i and $\sum_i \alpha_i = 1$. We implement a neural network based attention model. For each piece of memory m_i, we use a feed forward neural network to compute its semantic relatedness with the entity pair. The scoring function is calculated as follows:

$$g_i = \tanh(W_{word-att}[m_i; w_{eh}; w_{et}] + b_{word-att}) \tag{2}$$

where $W_{word-att} \in \mathbb{R}^{1 \times 3d}$ and $b_{word-att} \in \mathbb{R}^{1 \times 1}$. After obtaining $g_1, g_2, ...g_w$, we feed them to a $softmax$ function to calculate the final importance distribution $\alpha_1, \alpha_2, ...\alpha_w$.

$$\alpha_i = \frac{\exp(g_i)}{\sum_{j=1}^{w} \exp(g_j)} \tag{3}$$

One advantage of this model is that it could adaptively assign an importance score to each piece of memory m_i according to its semantic relatedness with the entity pair.

3.4 Relation Attention

Next, we explore the importance of all sentences for each relation and learn the dependencies between relations. The final representation of entity pair will be the composition of the sentence representations.

Selective Attention over Instances

In this section, for relation r_j-related model, we know that the final representation R_j depends on all sentences representations $X = \{x_1, x_2, ..., x_n\}$. Each sentence representation x_i contains information about whether entity pair (head,tail) contains relation r_j for input sentence s_i. Therefore, the vector R_j can be computed as the weighted sum of these sentence vector x_i:

$$R_j = \sum_{i=1}^{n} \beta_i x_i \tag{4}$$

where β_i is the weight of each sentence vector x_i. Then, we use a selective attention to de-emphasize the noisy sentence. Hence, β_i is further defined as:

$$\beta_i = \frac{\exp(z_i)}{\sum_{p=1}^{n} \exp(z_p)} \tag{5}$$

where z_i is a query-based function which scores how well the input sentence x_i and the predict relation r_j matches. We follow [Lin *et al.*, 2016] and select the bilinear form which achieves the best performance in different alternatives:

$$z_i = x_i A v_{r_j} \tag{6}$$

where A is a weight matrix, and v_{r_j} is the relation vector.

Selective Attention over Relations

Suppose there is a set R containing k relation representations for each entity pair, i.e., $R = \{R_1, R_2, ..., R_k\}$. To exploit the dependencies of all relations, we use selective attention to calculate the similarity between each relation. In this section, our attention function is the same as the instance-level attention model. The input is the output of the previous layer $\{R_1, R_2, ..., R_k\}$. The output R_j^* is computed as:

$$R_j^* = \sum_{i=1}^{k} \gamma_{ji} R_i \tag{7}$$

where γ_i is the similarity between relation R_j and R_i.

$$\gamma_i = \frac{\exp(h_i)}{\sum_{q=1}^{k} \exp(h_q)} \tag{8}$$

where h_i is referred as a similarity measure which scores how well the relation R_j and R_i correlates.

$$h_i = R_i B R_j \tag{9}$$

where B is a weight matrix.

3.5 Distant Supervised Relation Extraction

We regard the output vector $\{R_1^*, R_2^*, ..., R_k^*\}$ in last layer of relation-level memory network as the feature, and feed each one to a binary classifier. Therefore the confidence scores for each relation r_i can be calculated as:

$$o_i = W_i R_i^* + b_i \tag{10}$$

where matrix $W_i \in \mathbb{R}^{d \times 2}$ is the collection of weight vectors for each label and $b_i \in \mathbb{R}^2$ is a bias. Afterwards, we apply logistic function on each element of the score vector o_i to calculate the probability of each relation:

$$p(i|M, \theta) = \frac{1}{1 + exp(-o_i)} \tag{11}$$

where M denotes the set of aligned sentences, and $i \in \{1, 2, ..., k\}$, k is the number of relation labels. A binary label

vector **y** is used to indicate the set of true relations holding between the entity pair, where 1 means a true relation in the set, and 0 otherwise. Following this setting, we design a loss function for multi-label modeling:

$$loss = -\sum_{i=1}^{k} y_i log(p_i) + (1 - y_i) log(1 - p_i) \qquad (12)$$

where $y_i \in \{0, 1\}$ is the true value on label i. We use back propagation to calculate the gradients of all the parameters, and update them with Adadelta [Zeiler, 2012]. Dropout [Srivastava *et al.*, 2014] is also employed on formula (10) for regularization. We randomize other parameters with uniform distribution $U(-0.01, 0.01)$. We clamp the word embeddings with 50-dimensional vectors, which is the same as [Jiang *et al.*, 2016].

4 Experiment

We describe experimental settings and report empirical results in this section.

4.1 Experimental Setting

We conduct experiments on the basis of NYT10, a dataset developed by [Riedel *et al.*, 2010] and then widely used in distant supervised relation extraction [Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2012]. This dataset was generated by aligning Freebase relations with the New York Times (NYT) corpus, with sentences from the years of 2005 and 2006 used for training and sentences from 2007 used for testing. Statistics of the datasets are given in Table 1. It is worth noting that we follow [Jiang *et al.*, 2016] and use a filtered version of NYT10 released by them. The new version removes some relations which have very small number of instances.

Dataset	Sentences	Pos EPs	Neg EPs	relations
Training	112,941	4,266	61,460	26
Testing	152,416	1,732	91,842	26

Table 1: Statistics of the filtered NYT10 dataset, where EP denotes entity pair.

Following previous work [Lin *et al.*, 2016], we evaluate our method in the held-out evaluation. The held-out evaluation only compares the extracted relation instances, it gives a rough measure of precision without requiring expensive human evaluation. We evaluate the performance of each model with Precision-Recall curve and P@N metric.

4.2 Comparison with Existing Methods

We compare our approach with three traditional feature-based methods and two popular neural-based methods.

Feature-based methods

(1) **Mintz**: [Mintz *et al.*, 2009] proposed distant supervision paradigm and developed a multi-class logistic regression for classification. (2) **Multir** is a multi-instance learning method that was proposed by [Hoffmann *et al.*, 2011] with a deterministic "at-least-one" decision . (3) **MIML** [Surdeanu *et al.*, 2012] is a multi-instance multi-label approach for distant supervision using a graph model.

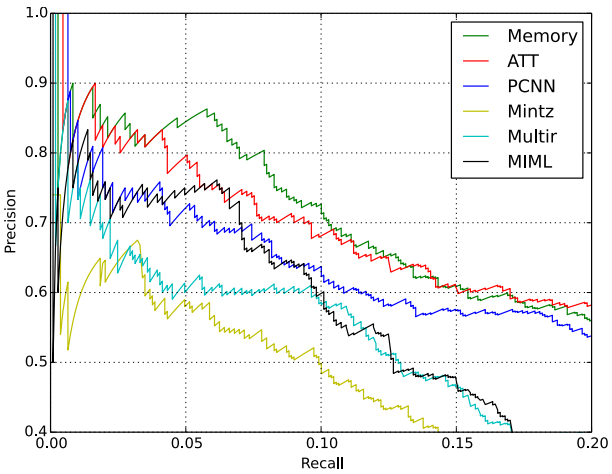

Figure 4: Precision-recall curves of various methods.

Neural-based methods

(1) **PCNN** [Zeng *et al.*, 2015] is a convolutional neural network based method for relation extraction. This method models overlapping relations by combining sentence-level relation extraction features into entity-pair-level results. (2) **ATT**: [Lin *et al.*, 2016] pointed out that distant supervision suffers from the entity pair wrong labeling problem. They developed a sentence-level attention model which can dynamically reduce the weights of those noisy instances and achieves state-of-the-art results.

Figure 4 shows the resulting precision-recall curve in the most concerned area. Our model is abbreviated to DMN, which contains a 6-hops word-level memory network and a two layer relation-level memory network. We can find that neural-based methods are extremely strong performer and substantially outperforms feature-based methods, demonstrating that the error propagation brought by NLP tools will hurt the performance of relation extraction. Among the three neural models, DMN and ATT perform better than PCNN, which indicates that taking the sentencelevel selective attention into account is helpful. In addition, we can see that DMN provides superior performance to all other methods by a wide margin, at least between 0 and 0.1 recall.

Table 2 further presents the results using P@N metric. In accordance with our observation in precision-recall curve, DMN is still the winner at most of the entire P@N levels. Another conclusion is that neural network methods are also good at predicting top-ranked results compared with traditional feature-based methods. This is probably caused by the

	Top 100	Top 200	Top 500	Average
Mintz	0.77	0.71	0.55	0.676
Multir	0.83	0.74	0.59	0.720
MIML	0.85	0.75	0.61	0.737
PCNN	0.84	0.77	0.64	0.750
ATT	0.86	0.80	0.68	0.780
DMN	**0.89**	**0.82**	0.68	**0.797**

Table 2: Precision values for the top 100, top 200, and top 500 extracted relation instances.

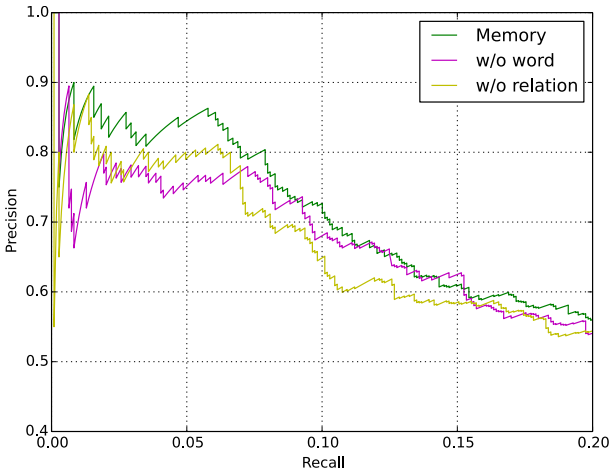

Figure 5: Effects of word-level and relation-level memory network.

imbalance of corpus and the wrong labeling problem that prevent it from reaching high precision when the recall keeps increasing.

4.3 Effect of word-level and relation-level Memory Network

The most important components in our model is the word-level memory network and the relation-level memory network. To demonstrate the effect of these two components, we further conduct ablation experiments, as shown in Figure 5. DMN is our full model with the two memory networks. "w/o word" is the system which removes the word memory part. In this way, the sentence representation is only learned by CNN. "w/o relation" is DMN without the relation memory part, using a sentence-level attention model instead. From the P-R curves of these three models, we can see that different memory networks have diverse emphases. When we incorporate the word-level memory network, most of the improvement resides in recall range [0.0, 0.1], but remains same trend in range [0.1, 0.2]. Compared with the word-level memory network, the relation memory brings improvement almost in all positions. Therefore, both memory networks play an important role in our model.

4.4 Effect of the Number of Hops

In this section, we further investigate the effect of the number of hops in word-level memory network. Results are summarized in Table 3, which lists the performance peak (highest F1 score) for each of the models. Among all of our models from 1 hop to 8 hops, we can observe that using more computational layers could generally lead to better performance, especially when the number of hops is less than three. When the number of hops exceeds 6, the experimental results become worse. We suggest that the reason might lie in the gradient vanishing problem as the memory network going deeper.

	Precision	Recall	F1
DMN-Word(1)	40.13	34.30	36.99
DMN-Word(2)	46.19	31.96	37.78
DMN-Word(3)	47.84	33.12	39.14
DMN-Word(4)	42.42	35.93	38.90
DMN-Word(5)	43.34	40.87	42.06
DMN-Word(6)	48.76	37.34	42.30
DMN-Word(7)	45.45	36.16	40.27
DMN-Word(8)	40.77	39.05	39.05

Table 3: Results of different word memory hops at the highest F1 point in the precision/recall curve on the dataset that contains groups with at least 10 mentions.

5 Related work

Distant supervised relation extraction is a fine-grained classification task in relation extraction, which aims at identifying the semantic relation of a sentence set expressed towards an entity pair [Mintz *et al.*, 2009]. Most of the existing works use machine learning algorithms, and build relation classifier from sentences with automatically annotated relation labels based on Knowledge Base. However, distant supervision inevitably accompanies with the wrong labeling problem. To reduce the impact of noisy data, [Riedel *et al.*, 2010] models distant supervision for relation extraction as a multi-instance single-label problem, which allows multiple mentions for the same tuple but disallows more than one label per object. [Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2012] adopt multi-instance multi-label learning in relation extraction.

Compared with feature-based methods, neural methods are attracting growing interest primarily due to their capacity of learning text representation from data without careful engineering of features, and capturing semantic relations between entity pair and context words in a more scalable way. [Zeng *et al.*, 2015] combines at-least-one multi-instance learning with neural network model to extract relations on distant supervision data. Specifically, a study [Lin *et al.*, 2016] has shown promising results on building sentence-level attention for dynamically calculate the weights of multiple instances. Despite the effectiveness of these approaches, these neural models (e.g. CNN) don't explicitly reveal the importance of context evidences with regard to an entity pair and they ignore the dependencies between relations. In this work, we develop two memory networks that explicitly encode the context importance towards a given entity pair and capture the dependencies between relation labels.

6 Conclusion

In this paper, we develop a novel neural model with two memory networks for distant supervised RE. In first word-level memory network, our model capture importances of context words and automatically model a semantic representation of sentences towards the entity pair. We also successfully devise relation-level memory network to capture the dependencies between relations and incorporating multi-instance multi-label learning. Experimental results show that the proposed approach offers significant improvements over comparable methods.

Acknowledgments

This work was supported by the National High Technology Development 863 Program of China (No. 2015AA015407), National Natural Science Foundation of China (No. 61632011 and No. 61370164).

References

[Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 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, 2011.

[Jiang *et al.*, 2016] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. pages 1471–1480, December 2016.

[Li *et al.*, 2012] Qi Li, Haibo Li, Heng Ji, Wen Wang, Jing Zheng, and Fei Huang. Joint bilingual name tagging for parallel corpora. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1727–1731. ACM, 2012.

[Lin *et al.*, 2016] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 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, 2009.

[Nguyen and Grishman, 2015a] Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. *Volume 2: Short Papers*, page 365, 2015.

[Nguyen and Grishman, 2015b] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 39–48, 2015.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[Rush *et al.*, 2015] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

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

[Sun *et al.*, 2015] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*, pages 1333–1339, 2015.

[Surdeanu *et al.*, 2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics, 2012.

[Tang *et al.*, 2016] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.

[Vu *et al.*, 2016] Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539, San Diego, California, June 2016. Association for Computational Linguistics.

[Weston *et al.*, 2014] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[Xu *et al.*, 2015] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 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)*, 2015.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[Zeng *et al.*, 2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344, 2014.

[Zeng *et al.*, 2015] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, pages 1753–1762, 2015.

[Zhang *et al.*, 2006] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *COLING-ACL'2006*, pages 825–832. Association for Computational Linguistics, 2006.