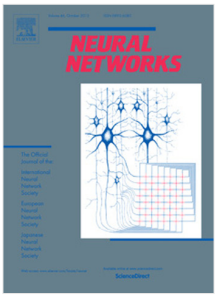
Accepted Manuscript



Distant supervision for neural relation extraction integrated with word attention and property features

Jianfeng Qu, Dantong Ouyang, Wen Hua, Yuxin Ye, Ximing Li

PII: S0893-6080(18)30006-6
DOI: https://doi.org/10.1016/j.neunet.2018.01.006
Reference: NN 3880

To appear in: *Neural Networks*

Received date : 16 June 2017
Revised date : 6 December 2017
Accepted date : 18 January 2018

Please cite this article as: Qu, J., Ouyang, D., Hua, W., Ye, Y., Li, X., Distant supervision for neural relation extraction integrated with word attention and property features. *Neural Networks* (2018), https://doi.org/10.1016/j.neunet.2018.01.006

***Title Page (With all author details listed)**

Author names and affiliations:

Jianfeng Qu[a,b], Dantong Ouyang[a,b], Wen Hua[c], Yuxin Ye[a,b,*], Ximing Li[a,b]

a College of Computer Science and Technology, Jilin University, Changchun 130012, China

b Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China

c School of Information Technology and Electrical Engineering, The University of Queensland, Australia

Corresponding author: Yuxin Ye yeyx@jlu.edu.cn +8618143095738 College of Computer Science and Technology, Jilin University, No. 2699 Qianjin Street, Jilin, Changchun 130012, China

Present/permanent address: College of Computer Science and Technology, Jilin University, No. 2699 Qianjin Street, Jilin, Changchun 130012, China

Distant Supervision for Neural Relation Extraction Integrated with Word Attention and Property Features

Jianfeng Qu[a,b], Dantong Ouyang[a,b], Wen Hua[c], Yuxin Ye[a,b,*], Ximing Li[a,b]

[a] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[b] *Key laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Minstry of Education, Changchun, 130012, China*
[c] *School of Information Technology and Electrical Engineering, The University of Queensland, Australia*

Abstract

Distant supervision for neural relation extraction is an efficient approach to extracting massive relations with reference to plain texts. However, the existing neural methods fail to capture the critical words in sentence encoding and meanwhile lack useful sentence information for some positive training instances. To address the above issues, we propose a novel neural relation extraction model. First, we develop a word-level attention mechanism to distinguish the importance of each individual word in a sentence, increasing the attention weights for those critical words. Second, we investigate the semantic information from word embeddings of target entities, which can be developed as a supplementary feature for the extractor. Experimental results show that our model outperforms previous state-of-the-art baselines.

Keywords: distant supervision, neural relation extraction, sentence encoding, word-level attention, supplementary feature

1. Introduction

Relation extraction (RE) aims to identify semantic relations between pairs of entities from plain texts. Recently, this task has attracted considerable attention [1] and has been a foundation for several important applications such as question answering (QA) [2] and knowledge graph construction [3].

*Corresponding author
Email address: yeyx@jlu.edu.cn (Yuxin Ye)

Table 1: Heuristic alignments between a KB and plain texts

Relation labels	President of (**Donald Trump**, **United States**)
Sentences in plain texts	S1. **Donald Trump** was elected the 45th President of the **United States**, after defeating Democratic candidate Hillary Clinton. (President of) S2. **Donald Trump** believes the **United States** has incredible potential and will go on to exceed anything that it has achieved in the past. (-) S3. **Donald Trump** said the **United States** needs to resume its surveillance of mosques, especially in New York City, where he says such surveillance has ceased. (-) . . .

Supervised RE systems require a large amount of human-labelled data to learn an extractor [4, 5], which is laborious and time-consuming. To get rid of human efforts, [6] proposes a distant supervision strategy which automatically generates training data through heuristic alignment between an existing knowledge base (KB) and plain texts. The alignment is based on the following assumption: if two entities from a KB participate in a relation, then all the sentences mentioning these two entities will express that relation. Table 1 shows an example of the alignment. It is apparent that the alignment will sometimes bring noise into the training data. For instance, S2 and S3 in Table 1 are wrongly labelled as training instances for the relation $President\_of$. [7, 8] ascribe the noisy data to a multi-instance problem, and adopts the at-least-one assumption (i.e., if two entities participate in a relation, then at least one sentence mentioning them might express that relation) to alleviate the noise. Furthermore, [9, 10] select multiple valid sentences by using sentence-level attention mechanism.

For the selected sentences in the training data, traditional feature-based methods [6, 7, 11] usually utilize lexical and syntactic features derived from natural language processing (NLP) tools for relation extraction, resulting in error propagation. To avoid the dependence on external tools, [8, 9, 10] apply piecewise convolutional neural networks (PCNNs) to straightforwardly encode sentence information. However, the existing neural methods for distant supervision are still confronted with two challenges.

Challenge 1 (Heterogeneous sentences): In sentence encoding, the existing neural models for relation extraction [5, 8, 9] are designed in a way that all words in a sentence contribute equally to predicting a relation between two target entities. However, this kind of treatment does not conform to heterogeneous sentences in plain texts. Take the

following sentence S as an example.

S: $Donald\ Trump$ was elected the 45th President of the $United\ States$, after defeating Democratic candidate Hillary Clinton.

Undoubtedly, the word "President" is of greater importance in predicting the relation $president\_of$ $(Donald\ Trump,\ United\ States)$, while the word "candidate" has little relevance with that relation. In general, the evidence for extracting a relation between an entity pair can simply consist of one or more key words instead of all words in a sentence. Therefore, the equal treatment of all words without any distinction will confuse the neural networks and degrade the performance of neural RE models.

Challenge 2 (Context-sparsity): Another challenge is the context-sparsity problem, namely distantly supervised relation extraction often suffers from the lack of useful sentence information. In distant supervision, the KB and plain texts used to generate training data do not have any internal links, because relational facts in the KB (e.g., Freebase) are mainly provided by user-submission rather than extracted from plain texts. Therefore, sometimes all the sentences in the given texts mentioning an entity pair do not express the relation that links those entities in the KB. Then there may be none valid sentences in some specific training data. In this case, sentence information cannot constitute the basis for predicting the given relation which, however, has been used in existing neural models as the only feature of positive instances during training process. Hence, it is necessary to seek supplementary information to resolve the context-sparsity problem.

Contributions. To address the first challenge, motivated by attention mechanism used in machine translation [12], we build a word-level attention-based module to encode sentences for distantly supervised RE. Specifically, our model computes the attention weight of each individual word in a sentence according to the intended relation between two target entities, and then aggregates them to form a vector representation for the sentence. In this way, our model will dynamically increase the weights of the critical words, while reducing the weights of the trivial words in a sentence. Ideally, the critical words will become the main components in the vector representations of sentences, so that the model can build a purified sentence encoding and facilitate the extractor to achieve accurate predictions.

To address the second challenge, we employ the property features derived from embeddings of entity pairs as a supplementary feature for the extractor. Previously, [13] has developed the embedding representation of words and found an interesting property that the difference between vector representations of two entities can reflect features about the relation between them. According to this property, in the study of knowledge graph completion [14, 15], researchers represent entities and relations in a common embedding space, and then predict new relational facts simply by vector computation between embeddings of entities and relations (i.e., $\boldsymbol{e_1} + \boldsymbol{r} \approx \boldsymbol{e_2}$ where the bold letters represent vectors). Inspired by these works, we leverage the property (i.e., $[\boldsymbol{e_2} - \boldsymbol{e_1}]$) as a supplementary feature for the relation extractor. In practice, property features enjoy the same status as sentence features and are designed to alleviate the context-sparsity problem in sentence features. Eventually, the evidence used for predicting the relation between an entity pair becomes a combination of the sentences mentioning that entity pair and the property embedded in the vector representations of those entities.

Our contributions in this paper can be summarized as follows:

- We build a word-level attention-based module for distantly supervised relation extraction. The module can allot larger attention weights to those critical words with respect to the potential relation between an entity pair, and construct a more purified representation for sentences.

- To the best of our knowledge, we are the first to propose the context-sparsity problem in distant supervision, and employ the property derived from word embeddings of entities as a supplementary feature for relation extractors to alleviate the problem. It is worth noting that during training process, word embeddings for entities will be regarded as normal parameters and updated according to the objective function, so that the property can be closer to the real relation characteristics.

- We conduct extensive experiments on real-world datasets, and the experimental results show that our model outperforms previous state-of-the-art baselines.

The remaining of the paper is organised as follows: in Section 2, we introduce

some preliminaries with an emphasis on the typical framework of neural relation extraction; we then elaborate on our proposed models and present the experimental results in Section 3 and Section 4 respectively; we summarize related work in the literature of relation extraction in Section 5, followed by a brief conclusion in Section 6.

2. Preliminaries

In this section, we briefly introduce some important concepts in distantly supervised relation extraction and summarize the notations used in this paper. Then we formally define the problem and introduce the framework of the state-of-the-art neural relation extraction model [9].

2.1. Problem definition

Definition 1 (Relation). *A relation $r(e_1,\ e_2)$ represents that entities e_1 and e_2 are related with the relation label r, e.g., $president\_of\ (Donald\,Trump,\ United\,States)$.*

Definition 2 (Bag). *A bag is a collection of sentences mentioning an entity pair. All the sentences that refer to entities e_1 and e_2 in a relation $r(e_1,\ e_2)$ constitute a bag, denoted as $(e_1,\ e_2)\ \{S_1,\ S_2,\ S_3, \cdots,\ S_n\}$, and r is the relation label of that bag. For example, S1, S2 and S3 in Table 1 constitute a bag for the entity pair $(Donald\,Trump,\ United\,States)$ and $President\_of$ is the relation label of that bag.*

The relation label r is known beforehand in the training process, while we need to predict r for a bag during testing. Hence, we formally define the problem of relation extraction as follows:

Definition 3 (Relation extraction). *Given a bag $(e_1,\ e_2)\ \{S_1,\ S_2,\ S_3, \cdots,\ S_n\}$, relation extraction aims to predict a corresponding relation label r for the entity pair $(e_1,\ e_2)$ and generate a relation $r(e_1,\ e_2)$.*

In Table 2, we summarize some important notations together with their explanations that will be adopted in the remaining of this paper. We will introduce them in detail later when used.

Table 2: A summary of notations used in this paper

$\boldsymbol{V}_S$	the distributed representation of the sentence S
$\boldsymbol{V}_{S_{con}}$	the convolutional encoding of the sentence S
$\boldsymbol{V}_{S_{wa}}$	the word-attention encoding of the sentence S
α_i	the attention weight of the word w_i
q_i	a query to score the relevance between the word w_i and the predicted relation r
$\boldsymbol{W}_d$	an intermediate matrix
$\boldsymbol{V}_{S_{bag}}$	the distributed representation of the bag of sentences
β_i	the weight of the sentence S_i
$\boldsymbol{e}_i$	the vector representation of the entity e_i

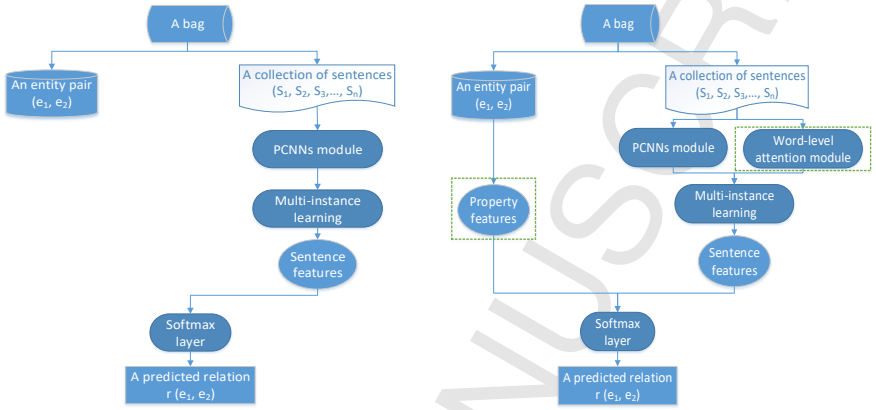
2.2. Framework overview

Figure 1(a) demonstrates the process of the current state-of-the-art neural relation extraction model introduced in [9]. As we can see, given a bag $(e_1,\ e_2)$ $\{S_1,\ S_2,\ S_3, \cdots,\ S_n\}$, the model simply utilizes the piecewise convolutional neural networks (PCNNs) module to extract sentence features. In order to deal with the noisy data problem, it further employs the multi-instance learning paradigm (i.e., sentence-level attention) to refine sentence features. Finally, a softmax layer is applied as the general extractor to predict the relation label for the given entity pair. Since the most important parts of this model are PCNNs and multi-instance learning, which also exist in our proposed model as depicted in Figure 1(b), we briefly introduce them in the following subsections.

2.2.1. PCNNs module

Given the sentence S, the PCNNs module transforms each raw word into a dense real-valued vector representation. To this end, word embeddings are employed to encode these words, which can be obtained by using the word2vec [1] tool. Additionally, the module needs position embeddings to inform the networks of the positions of the two target entities in the sentence S [5, 8]. Therefore, as shown in Figure 2, the input of the networks is an encoding matrix $\mathbb{R}^{d \times n}$. $d = dw + 2 * dp$, where dw and dp denote the dimensions of word embeddings and position embeddings respectively. n is the number of words in the sentence S.

[1] https://code.google.com/p/word2vec/



(a) State-of-the-art model [9] (b) Our proposed model

Figure 1: Framework overview of neural relation extraction models and the components in the dashed boxes of the right figure are the differences between these two models, which are also our main contributions.

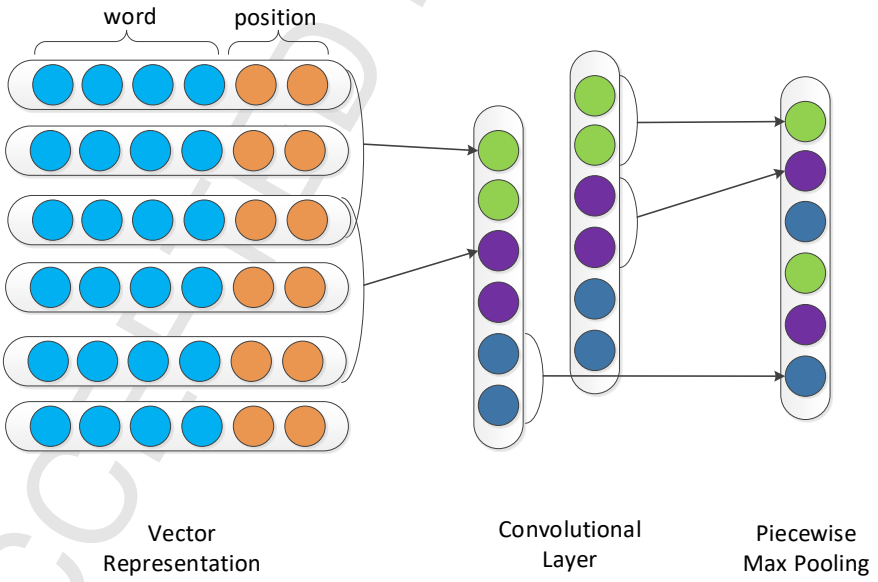


Figure 2: The architecture of PCNNs module, where the input is the vector representation of each word in a sentence. A convolutional layer together with the piecewise max pooling is utilized to extract every part of the features and generate the convolutional encoding of a sentence.

For the matrix $\mathbb{R}^{d\times n}$, a convolutional layer is adopted to extract every part of the features. Then piecewise max pooling generates the maximum value of each piece. The output of PCNNs module is the convolutional encoding of sentence S, denoted as $\boldsymbol{V}_{S_{con}}$ [8, 9].

2.2.2. Multi-instance learning for sentence features

During the training process of relation extraction models, for a relation $r(e_1, e_2)$, distantly supervised approaches regard all the sentences $\{S_1, S_2, S_3, \cdots, S_n\}$ that mention the entity pair as a training bag for the relation label r. Previous works assume that there exist one or more valid sentences in a training bag, and leverage multi-instance learning paradigm to resolve the mixture of valid and invalid sentences [7, 8, 9]. Among them, [9] uses a sentence-level attention-based method to select valid sentences in each training bag. Then the distributed representation of all the sentences in the bag can be computed as a weighted sum of these sentence vectors $\boldsymbol{V}_{S_i}$, namely

$$\boldsymbol{V}_{S_{bag}} = \sum_i \beta_i \boldsymbol{V}_{S_i} \tag{1}$$

where $\boldsymbol{V}_{S_{bag}} \in \mathbb{R}^{3nc}$ denotes the distributed representation of the bag of sentences and β_i denotes the weight of each sentence S_i. For conciseness, we refer readers to [9] for more details of the multi-instance learning method.

3. The Proposed Model

Figure 1(b) displays the framework of our proposed model for relation extraction. Our model differs from the current state-of-the-art method in two aspects (dashed boxes in Figure 1(b)), which are also the main contributions of this work. In particular, we propose a word-level attention module and property features respectively, to resolve the two challenges discussed in Section 1, namely heterogeneous sentences and context-sparsity.

Our model works as follows: given a bag $(e_1,\ e_2)\{S_1,\ S_2,\ S_3, \cdots,\ S_n\}$, the features of that bag are mainly categorized into two parts, i.e., sentence features and property features. To obtain sentence features, we propose a word-level attention module which is responsible for capturing the key words in each sentence for representing

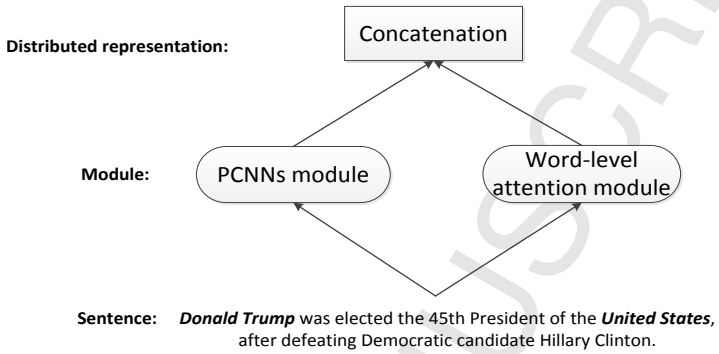


Figure 3: The architecture of our sentence encoder: two main components constitute the final distributed representation of a sentence and each component is designed to extract specific information in the sentence (i,e., PCNNs module extracts every local information while word-level attention module extracts key words).

the relations between entity pairs. Then the existing PCNNs module, together with the word-level attention module, extracts sentence features. Afterwards, we adopt the existing multi-instance learning paradigm (i.e., sentence-level attention) to optimize sentence features at the bag level. In terms of property features, we obtain the semantic information from word embeddings by formulating embeddings of e_1 and e_2. Finally, the combination of sentence features and property features constitutes the entire features for the softmax layer to generate the relation label r of the given bag $(e_1,\ e_2)\{S_1,\ S_2,\ S_3, \cdots,\ S_n\}$. In the following subsections, we will elaborate on the two contributions of this paper, i.e., word-level attention module and property features.

3.1. Sentence features

As shown in Figure 3, our encoder for sentences is composed of the PCNNs module and the word-level attention module. The output of these two modules will be concatenated to form the distributed representations of sentences. We denote the distributed representation of the sentence S as follows:

$$\boldsymbol{V}_S = [\boldsymbol{V}_{S_{con}}; \boldsymbol{V}_{S_{wa}}] \tag{2}$$

where $\boldsymbol{V}_{S_{con}}$ is the convolutional encoding of S, $\boldsymbol{V}_{S_{wa}}$ is the word-attention encoding of S, and $[\boldsymbol{V}_{S_{con}}; \boldsymbol{V}_{S_{wa}}]$ represents the vertical concatenation of $\boldsymbol{V}_{S_{con}}$ and $\boldsymbol{V}_{S_{wa}}$. Since $\boldsymbol{V}_{S_{con}}$ can be obtained by existing PCNNs module (described in Section 2.2.1), we introduce our novel word-level attention encoding for sentences in the following.

3.1.1. Word-level attention module

In relation extraction, the complexity and the heterogeneity of sentences have always been a thorny problem. Generally, not all words in a sentence play the equally important role in representing the relational fact between an entity pair of interest. For example, in the sentence "$Donald\ Trump$ was elected the 45th President of the $United\ States$, after defeating Democratic candidate Hillary Clinton.", the word "President" plays a more decisive role in extracting the relation $president\_of(Donald\ Trump, United\ States)$ than the other words, such as "elected", "defeating", and "candidate", etc. In order to capture these key words for predicting relations more accurately, we design a word-level attention-based mechanism in sentence encoding, which is depicted in Figure 4. Ideally, critical words will contribute more to sentence encoding so that the relation extractor can make accurate predictions without confusion.

Given a sentence S containing the entity pair (e_1, e_2) and a sequence of words $(w_1, w_2, w_3, \cdots, w_n)$ that constitute S, the vector representation of S, denoted as $\boldsymbol{V}_{S_{wa}}$, can be computed as a weighted sum of the corresponding word vectors $\boldsymbol{w_i}$, as shown in Figure 4. More formally,

$$\boldsymbol{V}_{S_{wa}} = \sum_i \alpha_i \boldsymbol{w}_i \tag{3}$$

where α_i denotes the attention weight of the word w_i, and $\sum_i \alpha_i = 1$. In sentence encoding, α_i directly determines the contribution of w_i to the final vector representation of S. During training process, α_i will be adjusted with respect to the current parameters so that the model can gradually increase the weights of key words, while reducing the weights of insignificant words. To this end, we further define α_i as:

$$\alpha_i = \frac{\exp(q_i)}{\sum_k \exp(q_k)} \tag{4}$$

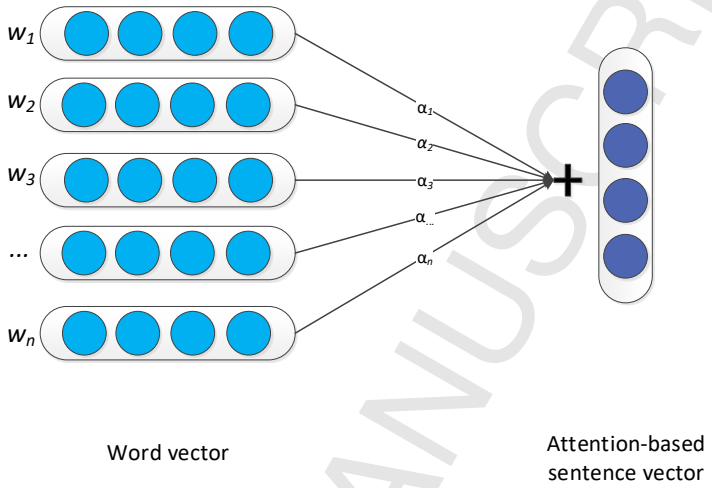


Figure 4: The architecture of word-level attention module, where $\boldsymbol{w}_i$ is the i-th word in a sentence and α_i indicates the attention weight for $\boldsymbol{w}_i$ assigned by our attention module.

where q_i can be regarded as a query to score the relevance between the word w_i and the predicted relation r.

However, the dilemma is that we do not know any information about the relation label that needs to be predicted in advance. Therefore, it is non-trivial to provide some cues about the relation between the target entity pair without using any contextual description.

TransE [14] is a landmark innovation which greatly facilitates the task of knowledge graph completion. Essentially, TransE places relations and entities into a common embedding space. Then the embeddings of the relation between two entities will be identified as a translation between the embeddings of two entities, which can be formalized as $\boldsymbol{e_1} + \boldsymbol{r} \approx \boldsymbol{e_2}$. In testing phase, given e_1 and r, TransE ranks all candidate entities e_2 based on the result of $||\boldsymbol{e_1} + \boldsymbol{r} - \boldsymbol{e_2}||^2$. Later works [15, 16] also follow this setting and achieve a remarkable improvement in knowledge graph completion. Inspired by these works, we use $[\boldsymbol{e}_2 - \boldsymbol{e}_1]$ to approximately represent the vector of the relation $\boldsymbol{r}$ to be predicted. The vector representation of e_1 and e_2 can be easily ac-

quired using existing word embedding methods. Then we give the following formula to compute q_i:

$$q_i = \boldsymbol{W}_d([\boldsymbol{w}_i; \boldsymbol{r}]) \tag{5}$$

where $[\boldsymbol{w}_i; \boldsymbol{r}] \in \mathbb{R}^{2dw \times 1}$ denotes a vertical concatenation between the word $\boldsymbol{w}_i$ and the relation $\boldsymbol{r}$. $\boldsymbol{W}_d \in \mathbb{R}^{1 \times 2dw}$ is an intermediate matrix that links the embedding representations of w_i and r to the relevance value between them, and formulates a linear relationship between q_i and $[\boldsymbol{w}_i; \boldsymbol{r}]$ [2].

3.2. Property features

Sentence features are the main repository for the task of relation extraction. Unfortunately, in distantly supervised methods, sole sentence features are not enough to learn a good relation extractor. As described in Section 1, sentence information often suffers from the context-sparsity problem, that is, none of the sentences in some training bags express the relation labels of those bags. In this case, sentence features will mislead the extractor during training process, degrading the final performance of the extractor. To address this issue, we try to develop another type of features, i.e., property features, and integrate them with sentence features to form a complete feature vector for the extractor.

Previously, [13] noticed that there exists an interesting property in word embeddings: $\boldsymbol{V}$("Madrid")$-\boldsymbol{V}$("Spain") is close to $\boldsymbol{V}$("Paris")$-\boldsymbol{V}$("France"). Inspired by this observation, in the task of knowledge graph completion, [14, 15] regard the relation r as a translation from the head entity e_1 to the tail entity e_2 (i.e., $\boldsymbol{e}_1 + \boldsymbol{r} \approx \boldsymbol{e}_2$) and learn embeddings for them (i.e., entities and relations) using margin-based ranking criterion. Finally, their model predicts the relations between two target entities simply by vector computation of the corresponding embedding representations, which achieves

[2] When speculating a new relationship, we generally tend to guess the simplest relation for an unknown relationship (similar to Occam's Razor). In our experiments, we found that the simple linear relationship can achieve the best results. Especially in the case study section, the proposed method is of the excellent ability in identifying the importance of each word and selecting those critical words that play a key role in representing relations. Therefore, we argue that the linear model can preserve semantic information embedded in words and provide enough capacity to serve as the attention module.

good performance. These researches have sufficiently demonstrated that the difference between word embeddings of an entity pair can reflect semantic information about the relation between them. Hence, we believe that this property can also provide effective evidence for the task of relation extraction.

Here, we take advantage of word embeddings for the entity pair independently and formulate them as a supplementary feature for the training bag. Specifically, given a bag labeled by $r(e_1, e_2)$, we use the difference vector of the entity pair $[\boldsymbol{e}_2 - \boldsymbol{e}_1]$ as one part of the features for this bag, which enjoys the same importance as the distributed representation of the sentences in the bag $\boldsymbol{V}_{S_{bag}}$. Finally, the combined features of the bag can be calculated as:

$$\boldsymbol{V}_{bag} = [\boldsymbol{e}_2 - \boldsymbol{e}_1; \boldsymbol{V}_{S_{bag}}] \tag{6}$$

where $\boldsymbol{V}_{bag} \in \mathbb{R}^{dw+3nc}$ and $[\boldsymbol{e}_2 - \boldsymbol{e}_1; \boldsymbol{V}_{S_{bag}}]$ is the concatenation of $[\boldsymbol{e}_2 - \boldsymbol{e}_1]$ and $\boldsymbol{V}_{S_{bag}}$. Obviously, $\boldsymbol{V}_{bag}$ is a combined representation of property information and sentence information. It is worth noting that we train the word2vec tool on New York Times beforehand to obtain the initial embeddings for entities. During training process of relation extraction, the embeddings of entities will be treated as normal parameters and updated according to the objective function, so that $[\boldsymbol{e}_2 - \boldsymbol{e}_1]$ can be closer to the real relation characteristics [3].

In practice, property features can not only alleviate the influence of the context-sparsity problem during training process, but also enable the extractor to make accurate predictions by considering multiple information in testing phase.

3.3. Objective function and optimization

Given a bag $T = (e_1,\ e_2)\ \{S_1,\ S_2,\ S_3, \cdots,\ S_n\}$, we define the conditional probability $p(r|T, \theta)$ through a softmax operation to compute the confidence of each possible relation:

[3] The initial word embeddings are obtained by training word2vec on New York Times and the internal mechanism in such training is based on the co-occurrence of words in a large number of texts. Therefore, these embeddings cannot sufficiently include semantic information that entities should have in the task of relation extraction. To improve the embedding representation, we regard them as normal parameters and update them during training process of the entire model.

$$p(r|T,\theta) = \frac{\exp(o^r)}{\sum_{k=1}^{nr} \exp(o^k)} \tag{7}$$

where nr is the total number of possible relations and $\boldsymbol{o} = \boldsymbol{M} \cdot (\boldsymbol{V}_{bag} \circ \boldsymbol{D}) + \boldsymbol{d}$. $\boldsymbol{M} \in \mathbb{R}^{nr \times (dw+3nc)}$ is a transform matrix from features to relations and each value in $\boldsymbol{M}$ represents the weight of the corresponding feature for predicting a specific relation label. $\boldsymbol{d} \in \mathbb{R}^{nr}$ is a bias vector, and $\boldsymbol{D}$ is a dropout vector of Bernoulli random variables with probability p used for regularization [17, 18] [4]. o^r represents the confidence that the bag T expresses the relation r.

Finally, we define the objective function for relation extraction using cross-entropy at the bag level as follows:

$$J(\theta) = \sum_{i=1}^{|T|} log\, p(Y_i|T_i,\theta) \tag{8}$$

where T_i is the i-th bag in the training data, Y_i is a possible relation label instantiated in T_i, and $|T|$ denotes the total number of bags. The parameter θ is a collection of all the parameters in the proposed model, including $\boldsymbol{M}$, $\boldsymbol{d}$, $\boldsymbol{W_d}$, $\boldsymbol{w}_i$, $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and the parameters in the PCNNs module. Similar to [8, 9], we employ the stochastic gradient descent (SGD) algorithm over mini-batches B to maximize the objective function.

4. Experiments

In this section, we empirically conduct comparative experiments to demonstrate the effect of the proposed method. For this purpose, we first introduce the dataset and the evaluation metrics used in the experiments. Then we describe some details about the model implementation. Finally, we present the experimental results along with some discussions.

[4] The main function of dropout is to force a neuron to work with other randomly selected neurons. Then the model can reduce joint adaptability among nodes in neurons and enhance the generation ability.

4.1. Dataset and evaluation metrics

We evaluate our model on a widely used dataset [5], which is developed by [7] and has also been used by [9, 10, 19]. The data is generated by aligning Freebase with New York Times corpus (NYT). To find entities mentioned in texts, they use Stanford named entity recognizer [6] and treat consecutive mentions which share the same category as a single entity mention. The association between Freebase and NYT is built by performing a string match between entity mention phrases and canonical names of entities in Freebase. The relations in Freebase are divided into two parts, one for training and the other for testing. Then the former is aligned to NYT in the year 2005-2006 and the later to NYT in the year 2007.

Following [6, 9], we adopt held-out evaluation. In testing phase, the precision and recall are calculated by comparing the predictions with the relational facts in Freebase. To sufficiently demonstrate the performance of each model, we evaluate them using various aspects of metrics, including precision/recall curves, the highest F1 value and P@N metrics.

4.2. Implementation details

In this paper, we train word embeddings on NYT corpus by using word2vec [7] tool in advance, and we concatenate consecutive words to represent an entity when the entity has multiple words. More importantly, the word embeddings obtained by word2vec will be used as the initial representation of words. We will treat them as parameters and modify them in training process, which can provide a better representation of words, especially entities, in relation extraction.

For our model, we tune all the parameters using three-fold validation. In detail, we use a grid search to determine the optimal parameters. The parameter settings are listed as follows: word dimension $dw = 50$; position dimension $dp = 5$; window size $l = 3$; convolutional filter $nc = 230$; batch size $B = 20$; dropout probability $p = 0.5$; learning rate $\lambda = 0.01$.

[5] Available at http://iesl.cs.umass.edu/riedel/ecml/

[6] Available at http://nlp.stanford.edu/software/CRF-NER.shtml

[7] Available https://code.google.com/p/word2vec/

4.3. Baselines

To evaluate the effect of our model, we empirically compare our methods (PCNN+ATT+WA, PCNN+ATT+PF, PCNN+ATT+WA+PF) with four strong baselines (Mintz, MultiR, MIML, PCNN+ATT). In order to guarantee a relatively fair comparison, for these baselines, we employ their publicly released source codes [8] and follow the parameter settings reported in their papers.

- *Mintz* is a traditional feature-based method proposed by [6].
- *MultiR* is a probabilistic graphic model which intends to resolve the multi-instance problem with overlapping relations in distant supervision [11].
- *MIML* is introduced in [19], which utilizes latent variables to alleviate the multi-instance multi-label (MIML) problem in the training data.
- *PCNN+ATT* is the current state-of-the-art neural relation extraction model [9]. It employs PCNNs module to realize automatic feature engineering. Moreover, the model utilizes a sentence-level attention-based mechanism to select valid sentences so that it can alleviate wrong labeling problem in distant supervision.

For our proposed model, we design **three** different kinds of methods, to illustrate the respective contributions of word-level attention and property features:

- *PCNN+ATT+WA* builds **W**ord-level **a**ttention-based module to encode sentences and integrates it to the previous method PCNN+ATT.
- *PCNN+ATT+PF* leverages the **p**roperty **f**eatures and integrates them with sentence features (without word-level attention module) to constitute an entire feature for the extractor.
- *PCNN+ATT+WA+PF* combines the above two methods.

[8] Mintz, MultiR and MIML are available at: http://nlp.stanford.edu/software/mimlre.shtml; PCNN+ATT is available at https://github.com/thunlp/NRE.

4.4. Experimental results

4.4.1. Held-out experiments

Precision/recall curves (Effect of the combined model). Figure 5 shows the precision/recall curves for our combined model and all the baselines. From Figure 5, we have the following observations:

(1) In most regions of the curves, PCNN+ATT+WA+PF yields the highest precision with the same recall. Moreover, PCNN+ATT+WA+PF obtains a constant and substantial improvement over PCNN+ATT, which currently has the best results reported on this dataset, with higher precision for the same recall, and higher overall recall (2.4%). We believe that the combination of word-attention module and property features constitutes more significant features and promotes the extractor to make accurate predictions.

(2) The feature-based methods (i.e., Mintz, MultiR and MIML) perform extremely worse than the neural models (i.e., PCNN+ATT, PCNN+ATT+WA+PF) in both precision and recall. The result illustrates that error propagation and accumulation in NLP tools are indeed a serious problem in relation extraction, degrading the effectiveness of relation extractors.

Precision/recall curves (Effects of word-level attention and property features respectively). To separately evaluate the effects of our two contributions, we present the precision/recall curves for PCNN+ATT+WA and PCNN+ATT+PF in Figure 6. From Figure 6, we can observe that:

(1) PCNN+ATT+WA generally outperforms PCNN+ATT with the relative improvement of 3.5%. Remarkably, when recall is low, PCNN+ATT+WA achieves much higher precision than all the other methods. These results demonstrate that only PCNNs module cannot capture the key words encoded in heterogenous sentences due to the identical treatments of words. And as expected, our word-level attention-based mechanism can indeed generate better sentence encoding for predicting relations by allocating greater weights to these critical words with respect to the desired relations. We believe that the integration of PCNNs module with word-attention module can facilitate the task of relation extraction to make accurate predictions. Later, we will give

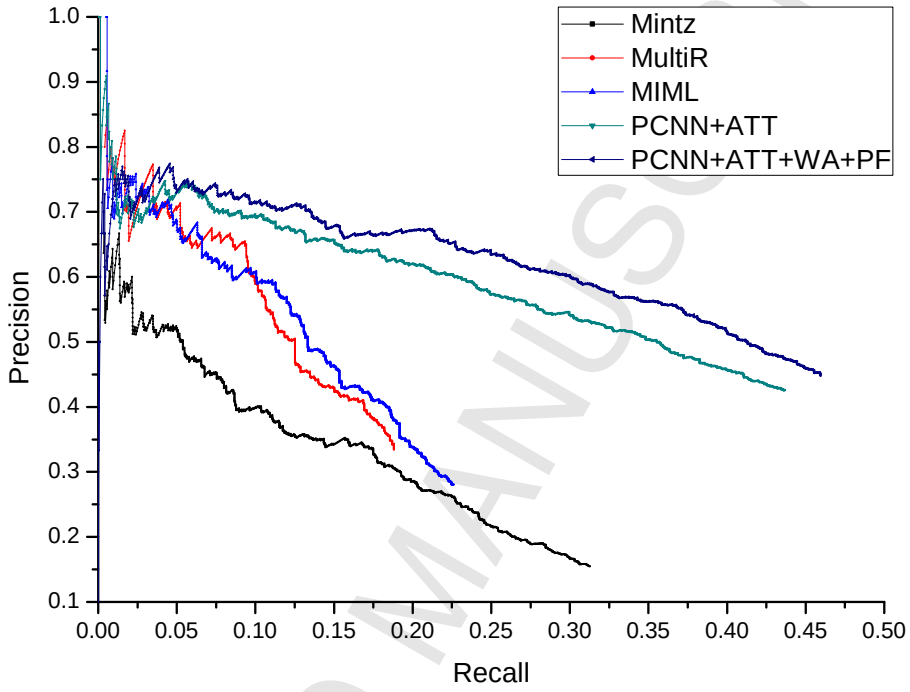


Figure 5: The precision/recall curves for the combined model and the baselines.

more detailed analysis of the attention mechanism.

(2) For most of the curves, especially when recall is between 0.15 and 0.35, PCNN+ATT+PF brings higher precision than PCNN+ATT. The difference can be as high as 6% around the middle of the curve. We conclude that the property features derived from embeddings for entities can provide additional semantic information, which reflects features about the relation linking two target entities and is beneficial to relation extraction. Then the combination of sentence features and property features can enhance the performance of the extractor by making full use of multiple effective evidence. In addition, PCNN+ATT only considers contextual information in plain texts, suffering from the context-sparsity problem in sentence features and degrading the performance of extractors in distantly supervised relation extraction. Nevertheless, when recall is larger than 0.35, PCNN+ATT+PF performs barely better than PCNN+ATT. We

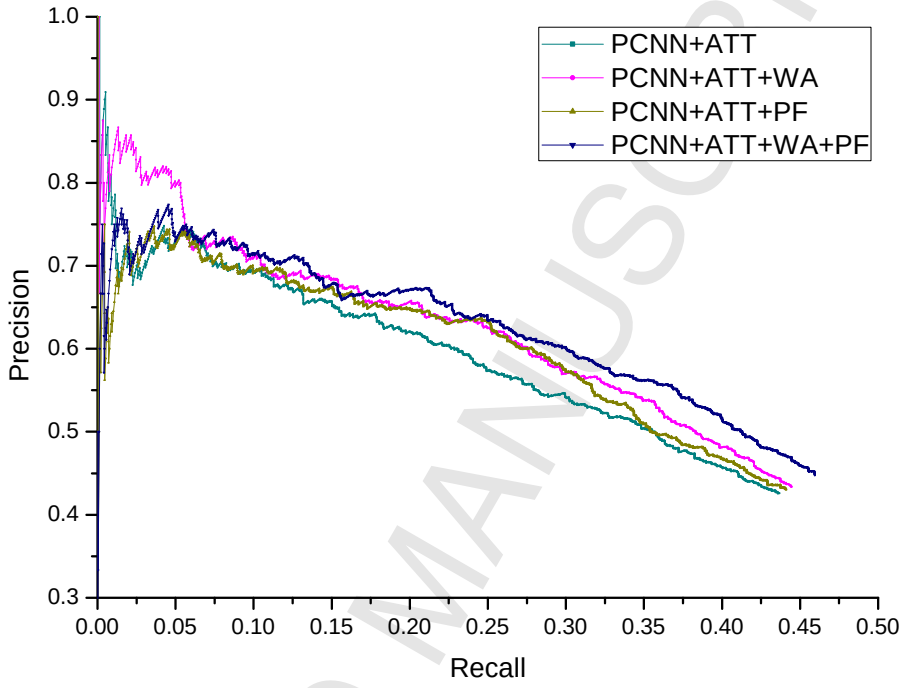


Figure 6: The precision/recall curves for the proposed methods.

believe that the phenomenon is caused by the weakness in sentence encoding. As the newly extracted instances increase, the weakness becomes more pronounced. Hence, the results further verify the necessity of the word-attention mechanism.

(3) Since PCNN+ATT+WA and PCNN+ATT+PF are proposed to solve two different problems in the existing methods, it is inappropriate to directly compare their performance. However, in most of the recall ranges, the combined model (i.e., PCNN+ATT+WA+PF) achieves higher precision than either of the separated models, which demonstrates that heterogeneous sentences and context-sparsity problems both exist in distantly supervised relation extraction and the combined model can simultaneously solve these two problems through word-level attention module and property features.

In some early experiments which we did not show in this paper, we have tested single word-attention module without PCNNs in sentence encoding, namely ATT+WA

Table 3: The highest F1 point in the precision/recall curves

Model	Precision(%)	Recall(%)	F1(%)
MintZ	26.17	22.67	24.29
MultiR	35.56	18.56	24.39
MIML	37.48	19.18	25.37
PCNN+ATT	42.91	43.44	43.17
PCNN+ATT+WA	46.89	41.69	44.14
PCNN+ATT+PF	43.61	43.74	43.67
PCNN+ATT+WA+PF	**46.94**	**44.46**	**45.67**

and ATT+WA+PF. Unfortunately, the results are far from satisfactory. The possible reason is that word-level attention mechanism emphasizes too much importance on individual words, while PCNNs module uses convolutional neural networks to master global sentence characteristics. Therefore, the proposed attention module mainly serves as a complementarity for the original encoding method, rather than a completely independent individual.

The highest F1 value. Following [19], we adopt F1 value to evaluate these methods. Table 3 reports the highest F1 [9] value of each method in the held-out evaluation. In Table 3, our methods (PCNN+ATT+WA, PCNN+ATT+PF and PCNN+ATT+WA+PF) achieve higher F1 values than all the baselines.

First, PCNN+ATT+WA obtains higher F1 than PCNN+ATT. It demonstrates that considering the relevance between each word in a sentence with the predicted relation r can bring better sentence encoding and boost the performance of the extractor. Second, PCNN+ATT+PF outperforms PCNN+ATT, which indicates that the difference of embeddings for an entity pair indeed contains semantic information about the relation between them. The combined features can give more evidence for predictions, making up for the lack of useful information in sentence features. Finally, PCNN+ATT+WA+PF achieves the highest F1 score, namely 2.5 points higher than PCNN+ATT, 1.53 points higher than PCNN+ATT+WA and 2.0 points higher than PCNN+ATT+PF. More importantly, with a better recall, PCNN+ATT+WA+PF obtains a precision that is over

[9] F1=2*precision*recall/(precision+recall)

Table 4: P@N for relation extraction

P@N(%)	300	500	1000	Mean
MintZ	44.98	39.69	33.60	39.42
MultiR	60.52	47.94	35.68	48.05
MIML	61.17	50.69	37.07	49.64
PCNN+ATT	69.00	63.80	55.00	62.60
PCNN+ATT+WA	69.67	66.20	57.80	64.56
PCNN+ATT+PF	69.67	65.40	58.00	64.36
PCNN+ATT+WA+PF	**71.00**	**66.80**	**59.50**	**65.77**

4 percentage points higher than that of PCNN+ATT. According to these results, we conclude that word-level attention-based mechanism and property features both are beneficial to neural relation extraction in distant supervision.

P@N metrics. Following [9], we also use P@N metrics to evaluate these models, which is showed in Table 4. To compute the precision of each model, we rank the predictions according to their confidence scores produced by these models. Then P@N is obtained by the precision of the top N. From Table 4, we can see that: (1) In P@300, P@500 and P@1000, PCNN+ATT+WA and PCNN+ATT+PF always get higher precision than all the previous methods, which demonstrates the effectiveness of our sentence encoding and property features, respectively. (2) In all testing sets, PCNN+ATT+WA+PF achieves the most outstanding performance. Remarkably, with the increasing size of the testing sets, compared with the state-of-the-art baseline (i.e., PCNN+ATT), PCNN+ATT+WA+PF acquires larger enhancement on precision (2.0% in P@300, 3.0% in P@500 and 4.5% in P@1000). Eventually, PCNN+ATT+WA+PF achieves the highest mean of precision which is 3.17 points higher than PCNN+ATT. Based on these, we conclude that our model can provide better sentence encoding and give more functional features for neural relation extraction.

4.4.2. Case study and discussion

Table 5 shows some examples of attention for words in testing data. The words with bold letters in a sentence are the target entities. For concise, we just list the words with relatively high weights allocated by our model from the corresponding sentences.

Table 5: Some examples of attention for words in the sentences from NYT

Relation	**Sentence**	**Words with high weight**
/business/person/company	the most visible and one of the most outspoken is **Vinod Khosla**, a founder of **Sun Microsystems** and now a partner at Khosla Ventures.	founder(0.071511)
/business/person/company	"we are very pleased with where it is today , " **Anne M. Mulcahy** , **Xerox** 's chief executive, said about PARC .	chief (0.084660) executive (0.068904)
/business/person/company	**Sherry Turkle** , a professor at the **Massachusetts Institute of technology** who studies the social aspects of technology, said that the participants on these sites are slipping into virtual worlds more easily than their parents or older siblings.	professor (0.086330)
/location/neighborhood/ neighborhood_of	last year, pacific retirement services, a nonprofit organization based in medford, ore. , began construction on the mirabella, a continuing-care community in the **South Lake Union** neighborhood of **Seattle**.	neighborhood (0.072716)
/people/person/nationality	officially a citoyen **Jonathan Littell**, the american author whose novel on the holocaust " the kindly ones " was last year's literary hit in **France**, has been granted french citizenship, agence france-presse reported yesterday	citizenship (0.057185)
/people/person/ place_of_birth	**Melvin Van Peebles** was born in **Chicago** in 1932	born (0.704648)
NA(None-relation)	"an unguided missile, " is how **Boutros Boutros-Ghali**, the former **United Nations** secretary general, once described him.	secretary (0.080033) general (0.068789)
NA	in **Chicago**, **Alfonso Soriano** led off the cubs' first inning with a home run, and it stood up for a 1-0 victory over Pittsburgh	home (0.052242)

From Table 5, we observe that the words with high weights often play a decisive role in predicting the relations. For example, "founder", "chief executive" and "professor" have higher relevance with the relation $/business/person/company$ than other words in their corresponding sentences. The results demonstrate that our attention mechanism can select critical words and make them occupy greater proportion in sentence encoding. On the other hand, since the weight of each word is computed by the relevance between the word and $[\boldsymbol{e_2} - \boldsymbol{e_1}]$, the results provide evidence that $[\boldsymbol{e_2} - \boldsymbol{e_1}]$ can approximately represent the relation $\boldsymbol{r}$ between e_1 and e_2. Further, we believe that $[\boldsymbol{e_2} - \boldsymbol{e_1}]$ can indeed serve as a supplement feature and provide informative messages for relation extraction.

In addition, we have inspected some misclassified examples generated by our model. The sentences listed in the last two rows of Table 5 are typical examples: (1) It's common sense for us that António Guterres is the current UN Secretary-General. However, in reality, there sometimes exist some descriptions about those past facts in plain texts. According to the relation types considered in this dataset, the relation between $Boutros\ Boutros\text{-}Ghali$ and $United\ Nations$ belongs to NA (here, the relation $/business/person/previous\_company$ doesn't exist). But our word-level attention strategy pays main attention to the words "secretary" and "general", while minor attention to the words "former", and wrongly predicts the entity pair with the relation $/business/person/company$. (2) Some other false positives come from terminologies. For example, in the last sentence, "home run" is a term in baseball. Nevertheless, our model misunderstands the word "home" as normal meaning and predicts a positive relation between $Alfonso\ Soriano$ and $Chicago$. (3) In term of false negatives, we analysed these wrong labels and found that the attention mechanism cannot handle too long sentences. For instance, the sentences containing entity pairs (e.g., ($Alaska,\ Prudhoe\ Bay$), ($Barry\ Diller,\ San\ Francisco$), ($Laura\ Lippman,\ Baltimore$)) have more than 50 words in each of them. Although our attention mechanism has successfully identified critical words, extensive useless words take up the majority of attention weights, distracting the model from a few key words. The above problems are infrequent in testing data, but we intend to resolve these problems in the follow-up study.

As for property features, we further have a post-hoc inspection of the results given by PCNN+ATT and PCNN+ATT+PF. In testing data, the sentences that mention the entity pairs in relations such as

$/location/neighborhood/neighborhood\_of(Montmartre,\ Paris)$,

$/location/location/contains(California,\ Hollywood)$,

and $/people/person/place\_of\_birth(Sharif\ Ahmed,\ Somalia)$

do not express the desired relation labels linking the pairs in the KB (like the context-sparsity problem in training data). Then it is understandable that PCNN+ATT certainly predicts these entity pairs with NA label. In contrast, PCNN+ATT+PF gives the desired results for these entity pairs, that is, the predictions are the same as the relational facts in the KB, respectively. The only possible explanation for these results is that property features (i.e., $[\boldsymbol{e_2} - \boldsymbol{e_1}]$) can indeed provide evidence for predicting the relations between entity pairs and make up for the lack of useful sentence information. Since property features can be effective in testing phase, we argue that the features are able to alleviate the context-sparsity problem in sentence information during training process.

4.4.3. Cross validation

In this part, we examine the robustness of the proposed model. To this end, we conduct 5-fold cross validation on Riedel's dataset[7]: the original training set and testing set are merged together, and then randomly partitioned into five subsets, one for testing and the remaining four for training. Table 6 records the overall performance of each model (the highest F1 value), where Pre is an abbreviation of Precision and Rec is an abbreviation of Recall.

In Table 6, we can find that: in each fold, the proposed three methods achieve higher F1 scores than PCNN+ATT, and finally obtain higher average F1 scores. Among them, PCNN+ATT+WA+PF is the most prominent, which gets an average F1 score with the relative improvement of 2.83% over PCNN+ATT. That is, faced with different training and testing samples, our model is often able to show the advantages in the sentence encoding as well as the multiple features. And these results reveal the robustness of our model. Additionally, in the aspect of stability, PCNN+ATT+WA+PF also achieves the

Table 6: 5-fold cross validation on Riedel's dataset (Std is an abbreviation of Standard Deviation).

	PCNN+ATT			**PCNN+ATT+WA**			**PCNN+ATT+PF**			**PCNN+ATT+WA+PF**		
	Pre(%)	Rec(%)	F1(%)	Pre(%)	Rec(%)	F1(%)	Pre(%)	Rec(%)	F1(%)	Pre(%)	Rec(%)	F1(%)
Fold 1	79.85	32.29	45.98	82.85	33.50	47.71	82.00	33.16	47.22	85.85	34.71	**49.43**
Fold 2	78.89	31.79	45.32	81.80	32.98	47.01	82.85	33.41	47.62	83.99	33.85	**48.25**
Fold 3	80.70	32.36	46.19	84.19	33.73	48.16	81.54	32.67	46.65	84.84	33.99	**48.54**
Fold 4	80.75	32.52	46.37	84.65	34.09	48.61	82.70	33.10	47.49	84.75	34.13	**48.66**
Fold 5	79.22	31.90	45.48	83.80	33.79	48.16	80.84	32.58	46.44	84.60	34.11	**48.61**
Mean±Std	79.88 ±0.84	32.17 ±0.31	45.87 ±0.45	83.46 ±1.14	33.62 ±0.41	47.93 ±0.61	81.99 ±0.83	32.98 ±0.35	47.08 ±0.52	84.81 ±0.67	34.16 ±0.33	**48.70 ±0.44**

best performance. Unfortunately, PCNN+ATT+WA and PCNN+ATT+PF are far from satisfactory. We argue that when the model both uses word-level attention mechanism and property features, the embedding representation for an entity pairs in each bag will be regarded as a parameter twice in one training iteration. Hence, the finally obtained embeddings for entity pairs can better reflect the features about the relations between entity pairs, and effectively serve as the metric for the attention module.

5. Related Work

Supervised relation extraction. Relation extraction is one of the most important research tasks in NLP. Many efforts based on supervised learning have been invested to boost the performance of relation extractors. [20, 21] employ kernel methods for relation extraction. Other classifiers, such as maximum entropy model [22] and conditional random fields [23], have also demonstrated the ability to achieve outstanding performance on domain-specific data. Recently, neural networks have been successfully applied to many NLP tasks [24]. To avoid hand-designed features, researchers have investigated the possibility of using neural networks to automatically learn features for relation extraction: recursive neural networks (RNNs) [25], convolutional neural network (CNN) [5] and long short-term memory (LSTM) [26]. However, supervised

methods rely entirely on manually annotated data, and cannot meet the demand of big data era.

Distantly supervised relation extraction. Distant supervision for relation extraction, firstly introduced by [6], automatically generates training data through heuristic alignment between a knowledge base and plain texts. Although distant supervision is an efficient way to scale relation extraction to a large number of relations, the basic assumption used in the alignment is so strong that it will inevitably bring wrong labelling problem. To alleviate noise, [7, 11, 19] build multi-instance learning paradigms. Specifically, [7] uses at-least-one assumption to resolve the problem. [11] builds a probabilistic graphic model and intends to resolve multi-instance with overlapping relations in distant supervision. [19] trains a Bayesian framework by expectation maximization (EM) algorithm. In addition, researchers notice that the incompleteness of the knowledge base (i.e., Freebase) will result in the false negative problem and design a latent-variable approach [27]. Later, considering automatic feature engineering, [8, 9] integrate multi-instance learning model with PCNNs to extract relations on distantly supervised data. Among them, [9] establishes sentence-level attention to select multiple valid sentences in each training bag, achieving the state-of-the-art performance.

The existing neural networks for relation extraction are designed in the way that all words in a sentence are of the same importance for predicting relations [5, 8, 9]. However, this type of treatment is not consistent with the reality. To address this shortcoming, [28] attempts to give these critical words more weights. Although this method achieves relatively high performance, there still exist some drawbacks which can be listed as follows: (1) the method obtains the weight of each word in a sentence by checking the relevance between the word and one single entity at a time. But the objective of relation extraction is to predict a relation between two entities. Therefore, just one entity may be insufficient to judge the importance of a word. In contrast, our model utilizes $[e_2 - e_1]$ to represent the relation r and computes the weight for each word by checking the relevance between the word and the relation r; (2) the method is designed for supervised relation extraction. As described above, supervised approaches suffer from the lack of training data, while distant supervision is a more promising

strategy.

Knowledge graph completion. [13] notices a valuable property in word embeddings: the difference between the vectors for an entity pair can reflect some features about the relation between them. Motivated by this discovery, [14, 15, 16] complete knowledge graph by regrading relation as a translation from the head entity e_1 to the tail entity e_2 ($\boldsymbol{e}_1 + \boldsymbol{r} \approx \boldsymbol{e}_2$). In their works, entities and relations are represented in a common space. The predictions given by their models are just based on additions and subtractions between the vectors for relations and entities without any contextual information.

In contrast to them, we do not need to explicitly denote the relation r with embedding representation in the task of relation extraction. To alleviate the context-sparsity problem in distant supervision, we straightforwardly attempt to employ the difference between entity pairs as a supplementary feature for neural relation extractors. Then the evidence used for predictions can be a combination of sentence information and property information. To the best of our knowledge, we are the first to develop the property derived from word embeddings for entity pairs to serve as features for neural relation extraction.

6. Conclusion

In this paper, we present a word-level attention-based mechanism and property features for neural relation extraction. The attention module can assign more attention weights to critical words and optimize encoding for sentences. The property features can leverage the property from word embeddings of entity pairs and provide more functional features for extracting relations, so as to alleviate the context-sparsity problem in distant supervision. Finally, we conduct comparative experiments to demonstrate the effectiveness of our model. The experimental results show that our model outperforms current state-of-the-art baselines.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) (grant number 61672261, 61502199, 61402196, 61272208); and Natural Sci-

ence Foundation of Zhejiang Province (grant number LY16F020004).

References

[1] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, in: Proceedings of IEEE, 2015, pp. 11–33.

[2] K. Xu, S. Reddy, Y. Feng, S. Huang, D. Zhao, Question answering on freebase via relation extraction and textual evidence, in: Proceedings of ACL, 2016, pp. 2326–2336.

[3] G. Weikum, M. Theobald, From information to knowledge: Harvesting entities and relationships from web sources, in: Proceedings of SIGMOD, 2010, pp. 65–76.

[4] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, Journal of Machine Learning Research 3 (1) (2003) 1083–1106.

[5] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: Proceedings of COLING, 2014, pp. 2335–2344.

[6] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of ACL, 2009, pp. 1003–1011.

[7] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2010, pp. 148–163.

[8] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: Proceedings of EMNLP, 2015, pp. 1753–1762.

[9] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, in: Proceedings of ACL, 2016, pp. 2124–2133.

[10] G. Ji, K. Liu, S. He, J. Zhao, Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: Proceedings of AAAI, 2017, pp. 3060–3066.

[11] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D. S. Weld, Knowledge-based weak supervision for information extraction of overlapping relations, in: Proceedings of ACL, 2011, pp. 541–550.

[12] M.-T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[14] A. Bordes, N. Usunier, A. Garcia-Duran, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.

[15] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of AAAI, 2015, pp. 2181–2187.

[16] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of ACL, 2015, pp. 687–696.

[17] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing coadaptation of feature detectors, Computer Science 3 (4) (2012) 212–223.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[19] M. Surdeanu, J. Tibshirani, R. Nallapati, C. D. Manning, Multi-instance multi-label learning for relation extraction, in: Proceedings of EMNLP, 2012, pp. 455–465.

[20] S. Zhao, R. Grishman, Extracting relations with integrated information using kernel methods, in: Proceedings of the 43rd annual meeting on association for computational linguistics, 2005, pp. 419–426.

[21] R. C. Bunescu, R. J. Mooney, Subsequence kernels for relation extraction, in: Advances in neural information processing systems, 2006, pp. 171–178.

[22] N. Kambhatla, Extracting relations with integrated information using kernel methods, in: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, 2004, p. 22.

[23] A. Culotta, A. McCallum, J. Betz, Integrating probabilistic extraction models and data mining to discover relations and patterns in text, in: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, 2006, pp. 296–303.

[24] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, J. Zhao, B. Xu, Self-taught convolutional neural networks for short text clustering, Neural Networks 88 (1) (2017) 22–31.

[25] R. Socher, B. Huval, C. D. Manning, A. Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: Proceedings of EMNLP, 2012, pp. 1201–1211.

[26] M. Miwa, M. Bansal, End-to-end relation extraction using lstms on sequences and tree structures, in: Proceedings of ACL, 2016, pp. 1105–1116.

[27] A. Ritter, L. Zettlemoyer, Mausam, O. Etzioni, Modeling missing data in distant supervision for information extraction, Transactions of the Association for Computational Linguistics 1 (2013) 367–378.

[28] L. Wang, Z. Cao, G. de Melo, Z. Liu, Attention-based convolutional neural network for semantic relation extraction, in: Proceedings of ACL, 2016, pp. 1298–1307.