Feature-level Attention based Sentence Encoding for Neural Relation Extraction

Longqi Dai, Bo Xu, and Hui Song*

School of Computer Science and Techology, Donghua University, Shanghai, China
2171743@mail.dhu.edu.cn, {xubo,songhui}@dhu.edu.cn

Abstract. Relation extraction is an important task in NLP for knowledge graph and question answering. Traditional relation extraction models simply concatenate all the features as neural network model input, ignoring the different contribution of the features to the semantic representation of entities relations. In this paper, we propose a feature-level attention model to encode sentences, which tries to reveal the different effects of features for relation prediction. In the experiments, we systematically studied the effects of three strategies of attention mechanisms, which demonstrates that scaled dot product attention is better than others. Our experiments on real-world dataset demonstrate that the proposed model achieves significant and consistent improvement in the relation extraction task compared with baselines.

Keywords: Relation Extraction · Feature-level Attention · Attention Strategies.

1 Introduction

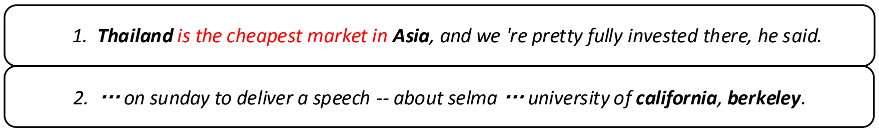
Relation extraction (RE), is defined as the task of extract relational facts from plain text. The goal of relational extraction is to extract relationships between entities mentioned in text, such as *LiveIn (person, location)* or *Founder (person, company)*. It is a crucial task in natural language processing (NLP) field, particularly for knowledge graph completion and question answering.

Researchers have added many extra features (e.g. part-of-speech, wordnet, named entity recognition, parse tree, etc.) beyond n-grams when utilizing traditional machine learning to perform relational extraction tasks [6,10], which has proven to be effective. In recent years, deep learning methods have been widely used for RE, that is, using neural networks to modeling relation extraction tasks. Neural relation extraction methods can be divided into two classes: (1) convolutional neural networks [15,26]. (2) sequence modeling: recurrent [23,28] and recursive [5,19] neural networks.

However, whether traditional machine learning or deep learning method, these models simply concatenate all the features involved [9,11,12] as the input representation of the model, without taking into account the different contribution of different features to the relation extraction task. As shown in Fig. 1, for the first sentence, the region features

* Correspondence author. This paper was sponsored by Shanghai Sailing Program No.19YF1402300, by the Initial Research Funds for Young Teachers of Donghua University No. 112-07-0053019

for words sequence (part of red color) clearly express the *Contains* relationship, but in the second sentence, the lexical feature and position feature give more cues to predict the relationship between entities. Therefore, in this paper, we proposed a feature-level attention model to encode sentence, which reveals the effects of features for relation extraction. The attention mechanism actively adjusts the weight of features based on context rather than simply concatenating multiple features directly.



1. ***Thailand*** *is the cheapest market in* ***Asia****, and we 're pretty fully invested there, he said.*

2. *··· on sunday to deliver a speech -- about selma ··· university of* ***california****,* ***berkeley****.*

Fig. 1. The triple of these examples is *Contains (location, location)*.

The contributions of this paper are summarized as follows:

- We proposed a feature-level attention model to encode sentence, which focuses on the contribution of different features to relation extraction, instead of the simple concatenation.
- To select the attention strategy that is more suitable for relation extraction, we systematically studied the effectiveness of the three score functions of attention mechanism, and found that the scaled dot product strategy achieves the best performance.
- In the experiments, we compared our feature-level attention model with other baselines of different granularity, and our model achieved the best results.

2 Related Work

2.1 Sentence Features for Relation Extraction

An important challenge in modeling relational extraction tasks is to design and select common, high-quality features. Many traditional machine learning approaches [6,10] described various useful features for relation extraction, such as words, entity type, mention level, overlap, dependency, parse tree, etc. However, these features are calculated based on existing NLP tools, so inevitably lead to error accumulation. Therefore, in recent years with deep learning methods being widely used in various fields of natural language processing, researchers [11,26] have attempted to use only the necessary basic features (usually word embedding feature and position feature) as input representations of neural network models, and gradually ignore artificially constructed features.

However, whether traditional machine learning or deep learning method, these models only simply concatenating all the features used, and then directly as the input representation of the model, without considering the contribution of different features to the relation extraction task is not equal. In this paper, we present a feature-level attention-based model that focuses on the contribution of different features to relation extraction.

2.2 Attention Mechanism on Relation Extraction

Bahdanau et.al. [1] proposed the attention mechanism in machine translation, which was later popularly in text summaries [18], image captioning [24], etc. and achieved great success. Besides, many formulas for calculating attention scores have been proposed. Common choices [13] include additive, multiplicative, multi-layer perceptron, hierarchical attention, self-attention, and more.

In addition, the use of attention mechanisms at different granularities is widely adopted by RE. Lin et al. [11] proposed a sentence-level attention-based model for instances selection to reduce the noise of distant supervision. Based on the research of [11], Liu et al. [12] and Jat et al. [9] proposed entity-pair level soft labeling method and word-level attention-based model for distant supervised relation extraction, respectively. In this paper, in order to extract the semantic relations in sentences more exactly, we propose a feature-level attention model for relation extraction.

2.3 Distant Supervision Relation Extraction

Supervised models [4] usually require large amounts of high-quality annotated data for relation extraction. To avoid the laborious and expensive task of manually building dataset, Mintz et al. [14] proposed a distant supervision approach for automatically generating adequate amounts of training data. However, distant supervision assumes that if two entities have a relationship in knowledge bases (KBs), then all sentences containing these two entities have a certain relationship, it inevitably suffers from the wrong labeling problem. To alleviate this problem and denoise, the multi-instance learning [17] framework is applied as a basic module in many researches works [2,8,11,21,25,26] of distant supervision. Our work continues these frameworks and try to improve the performance.

3 Overview

The neural relation extraction aims to predict the relation for the entity-pair via a neural network. In practical applications, obtaining a large amount of manually constructed training data is very expensive and cumbersome, distant supervision methods are popular latterly. Following Riedel et al. [17], Lin et al. [26], we utilize the multi-instance learning framework and instance selector to alleviate the wrong labelling problem of distant supervised relation extraction. In our experiments, we utilized the NYT10 dataset, which was automatically generated using the distant supervision paradigm [14].

In this section, we first introduce the basic notations and the features referred, and then present the overall framework of our approach for relation extraction, starting with notation.

3.1 Notations

Knowledge Graph. A Knowledge Graph is defined as $G = (V, E, F)$, where V, E, and F represent the collections of entities, relations, and facts, separately. For a relational

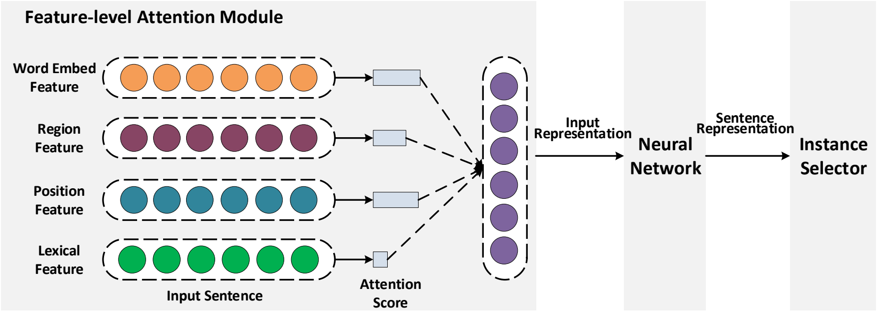


Fig. 2. The architecture of Feature-level Attention model.

fact $(h, r, t) \in F$ of a sentence, $h \in V$and $t \in V$ represent the head entity and tail entity of the sentence, and $r \in E$ denotes that the relation r in the entity pair (h, t).

Entity-Pair Bag. In multi-instance learning framework, all instances in a dataset are divided into multiple entity-pair bags $\{B_1, B_2, B_3, \dots\}$, where each bag B_i corresponds to multiple instances $\{s_1, s_2, s_3, \dots\}$ of a same entity-pair (h_i, t_i). For each instance s_i that contains multiple words, we denote that $s_i = \{w_1, w_2, w_3, \dots\}$.

3.2 Input Features

Word embedding feature is proposed by Hinton [7]. Given a sentence $s = w_1, w_2, \dots, w_n$, we adopt pretrained word embedding to transform each word w_i, denoted by $\boldsymbol{s}_w$.

Position feature is proposed by Zeng et al. [27], which aims to point out the relative distances from the word to head entity and tail entity in the sentence. Each word has two relative distances, denoted by $\boldsymbol{s}_{p1}$, $\boldsymbol{s}_{p2}$, separately.

Lexical feature. We using the existing NLP tools[1] to calculate the lexical features of the sentence, including part-of-speech tagging and named entity recognition, denoted by $\boldsymbol{s}_{lp}$, $\boldsymbol{s}_{ln}$, respectively.

Region feature. In relation extraction, input sentence is divided into three regions by its corresponding entity pair. We extend this clue to region features. For example, for the sentence "… the former *Pairs* home of the duke of westminster , a short walk from the tuileries gardens and the *Louvre*, is offering …", the word before *Pairs* in the sentence is in the left area, recorded as 0, the word between *Pairs* and *Louvre* is in the middle area, recorded as 1, and the word after *Louvre* is in the right area, recorded as 2. We further exploit the region feature as a component of the input representations of sentences, denoted by $\boldsymbol{s}_r$.

[1] http://www.nltk.org

3.3 Framework

As shown in Fig. 2, our model contains three modules, named *Feature-level Attention Module, Neural Network Encoder*, and *Instances Selector*. We describe them in the subsequent sections.

Feature-level Attention Module. In this module, we utilize the attention mechanism to calculate the weight of features. The weight of each feature represents the contribution of the feature to the semantic relationship of the sentence. This module will be described in detail in Section 4.1 below.

Neural Network Encoder. For the input representation computed by *Feature-level Attention Module*, we employ an extension convolutional neural network (PCNN) to obtain the sentence representation. More detail is shown in Section 4.2.

Instance Selector. When the entire sentences representation is learnt in the corresponding bag, we utilize selective attention paradigm to select the instances which really express the semantic relation. Please refer to Section 4.2 for details.

4 Methodology

Given an entity pair (h, t) and its corresponding bag B partitioned by multi-instance learning framework, the purpose of neural relation extraction model is to measure the conditional probability $p(r|B, \theta)$ of relation $r \in R$ via a neural network.

In this section, we first introduce our *Feature-level Attention* method applied to the input representation and then we employ the neural architecture: PCNN [26] as the *Neural Network Encoder* and selective attention [11] as the *Instance Selector*, described in detail in Section 4.2. Fig. 2 shows the architecture of our method for distant supervised relation extraction. The leftmost side is the attention features module we proposed, and the rest part is the basic neural architectures.

4.1 Feature-level Attention

Given a sentence $s = \{w_1, w_2, \dots, w_h, \dots, w_t, \dots\}$, it contains two entities (w_h, w_t), and the initial representation **x** is composed of four input features (i.e. word embed, region, position and lexical). We fixed the dimension of input features to d_s and utilized the attention mechanism to obtain the weight α of each feature. As described in the transformer [22], the attention mechanism can be described as mapping a query and a set of key-value pairs to an output. That is, the computation of the attention mechanism consists of three matrices (i.e. K, V, and Q), as shown in Fig. 3.

In calculation, the input component keys, values, and queries are all matrices, which are represented by K, V, and Q respectively. The output matrix **g** is computed as a weighted sum of values. The formalization of attention mechanisms is defined as:

$$\boldsymbol{e} = score\ function(K, Q) \tag{1}$$

$$\mathbf{g} = \sum softmax(\boldsymbol{e})V \tag{2}$$

where $\boldsymbol{e}$ is the attention score calculated using the scoring function, and **g** is the input representation after encoding with attention.

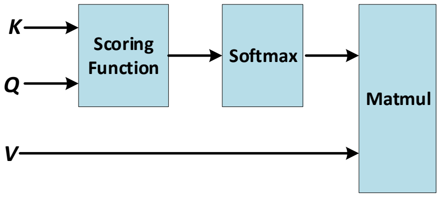


Fig. 3. Attention Mechanism.

While various existing scoring strategies [13,22] could be deployed in this setup, we explored these different strategies. Three scoring strategies are easily implemented in most common neural models for relation extraction.

Scaled Dot Product: This method adds a scaling factor $1/\sqrt{d_s}$ to prevent the softmax function pushed into small gradients region, which is a variant of dot product attention [22]. Formalized as:

$$score\ function(K, Q) = \frac{QK^T}{\sqrt{d_s}} \tag{3}$$

where $K = V = \mathbf{x}$, and following the translation-based knowledge graph method [3], we use the difference tensor of entity word embedding to represent the relationship (i.e. $Q = \boldsymbol{w}_h - \boldsymbol{w}_t$).

Additive Attention: The additive method introduces parameter matrices $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ that uses a feed-forward network instead of dot-product to compute attention scores [13], K and Q are identical to the dot product method.

$$score\ function(K, Q) = \boldsymbol{W}_1^T tanh(\boldsymbol{W}_2[Q, K]) \tag{4}$$

Self-Attention: For the general dot-product or additive method, they must employ the difference tensor of entities embedding to estimate the relationship Q, which includes noise. In self-attention, only the initial representation $\mathbf{x}$ of the sentence are involved to compute the attention score, defined as:

$$Q = \boldsymbol{W}_q\mathbf{x}, \;\; K = \boldsymbol{W}_k\mathbf{x}, \;\; V = \boldsymbol{W}_v\mathbf{x} \tag{5}$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ are parameter matrices, and the specific formula for attention score can adopt additive or dot-product method.

4.2 Neural Architectures

Neural Network Encoder: We fusion all the input features to compute the input representation **g**=$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3 \ldots, \boldsymbol{w}_n\}, \boldsymbol{w}_i \in R^{d_s}$(see section 4.1), then we adopt an extension convolutional neural network PCNN [26] to encode input representations into sentence embeddings.

For the convolution operation, the window size of the convolution kernel is defined as l, then the vector of the concatenation of words within the i-th window ($\boldsymbol{q}_i \in R^{d_s \times l}$) can be defined as:

$$\boldsymbol{q}_i = \boldsymbol{w}_{i:i+l-1}; \; (1 \leq i \leq n - l + 1) \tag{6}$$

We further define the convolutional matrix is $\boldsymbol{W}_c \in R^{d_c \times (d_s \times l)}$ and bias vector is $\boldsymbol{b} \in R$, where d_c is the sentence embedding size. The output for the i-th filter $\boldsymbol{c}_i = [\boldsymbol{W}_c \boldsymbol{q} + \boldsymbol{b}]_i$. Afterwards, a piecewise max-pooling is used to divide the convolution filter $\boldsymbol{c}_i$ into three regions $\{\boldsymbol{c}_{i,1}, \boldsymbol{c}_{i,2}, \boldsymbol{c}_{i,3}\}$ by two entities. The final sentence embedding $\boldsymbol{s}$ is defined as:

$$\boldsymbol{s}_i = [max(\boldsymbol{c}_{i,1}), max(\boldsymbol{c}_{i,2}), max(\boldsymbol{c}_{i,3})] \tag{7}$$

Instance Selector: Given the entity pair and its bag of instances $B = \{s_1, s_2, \ldots, s_t\}$, we obtain the instance embeddings $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_t\}$ using encoder layer. Instance selector aims to compute the textual relation representation $\mathbf{u}$ over all the instances in the bag, we use selective attention schema [11] to measure the attention score θ_i for instances in the bag.

$$\mathbf{u} = \sum_i \theta_i \boldsymbol{s}_i, \quad \theta_i = \frac{\exp(\boldsymbol{s}_i \boldsymbol{A} \boldsymbol{q}_r)}{\sum_z \exp(\boldsymbol{s}_z \boldsymbol{A} \boldsymbol{q}_r)} \tag{8}$$

where $\boldsymbol{A}$ is the weight matrix and $\boldsymbol{q}_r$ is the relation query vector associated with relation $r \in R$.

4.3 Optimization and Implementation Details

Here we introduce the learning and optimization details for our feature attention model.

For the output $\mathbf{u}$ of the *Instances Selector* module, we adopt a softmax layer to measure the conditional probability $p(r|B, \theta)$,

$$p(r|B, \theta) = \frac{\exp(\mathbf{o}_r)}{\sum_{z \in R} \exp(\mathbf{o}_z)}, \quad \mathbf{o} = \mathbf{M}\mathbf{u} + \mathbf{d} \tag{9}$$

where $\mathbf{o}$ is the output score of all relation types, $\mathbf{M}$ is the representation matrix and $\mathbf{d}$ is bias vectors.

Given the set of entity-pair bags $\pi = \{B_1, B_2, \ldots\}$ and corresponding label set $\{r_1, r_2, \ldots\}$, the loss function is given as,

$$J(\theta) = -\sum_{i=1}^{|\pi|} \log p(r_i|B_i, \theta) \tag{10}$$

For the implementation, we apply dropout regularization [20] on the output layer of our models to guard against overfitting.

5 Experiments

5.1 Dataset and Evaluation Metrics

We performed experiments on the NYT10 dataset and adopted cross-validation to evaluate our feature attention method. The dataset[2] is constructed by aligning Freebase triple

[2] http://iesl.cs.umass.edu/riedel/ecml/

with the New York Times (NYT) corpus, which is developed by Riedel et.al. [17], where sentences from the year 2005-2006 are used for building the training set and from the year 2007 for the testing set.

NYT10 dataset contains 53 relations including an NA relation that indicates that there is no relation in the instance, and the dataset is commonly used in related works. The training set contains 522,611 sentences, 281,270 entity pairs and 18,252 relational facts. The testing data contains 172,448 sentences, 96,678 entity pairs and 1,950 relational facts.

5.2 Baselines

To evaluate our approach, we compared the following baselines:

Mintz [14] is a logistic regression model for distant supervision paradigm.

MultiR is proposed by Hoffmann et al. [8], which is a probabilistic, graphical model for multi-instance learning.

MIML [21] proposed a multi-instance multi-label model for distance supervision.

PCNN [26] is an extension to convolution neural network, which employ a piecewise max-pooling layer for instance embeddings.

PCNN+ATT [11] proposed a sentence-level attention mechanism for instance selection.

PCNN+ATT+SL [12] employs an entity-pair level soft-label method to dynamically reduce the noise of the wrong annotations.

BGWA [9] is a word-level attention approach based on Bi-GRU.

AFPCNN is proposed by us, using extra features and feature-level attention method to encode sentences. More details in Section 4.1.

5.3 Parameter Settings

For the experiment, we utilized glove [16] that trained the word embedding on New York Times Corpus, which has $d_s = 50$ dimensions. We compared the score function of attention module among self-attention, additive, scaled dot product, and the best one is scaled dot product. For model parameters, we empirically set the batch size $B_s = 160$, the learning rate $\lambda = 0.2$, decay rate $\epsilon = 10^{-9}$, the window size $l = 3$ of convolution kernel, and the sentence feature maps $d_c = 230$. In training, we employed the dropout strategy to guard against overfitting and take SGD as the back-propagation algorithm.

5.4 Effect of Feature-level Attention

To demonstrate the validity of the proposed approach, we compare it with the previous baselines (See section 5.2), the Precision-Recall curves is shown in Fig. 4. To measure the contribution of each feature to the relation extraction, we set the dimensions of all features to 50 and using scaled dot product as the score function of attention. Overall, our models achieved higher AUC values and F1 scores on the NYT10 dataset. More detailed P@N metric with N = $\{100, 200, 300\}$ and the Area Under the Precision-Recall Curves are shown in Table 1.

In Fig. 5, we explore the experimental results from the perspective of model granularity. We compare our feature-level relation extraction model (**AFPCNN**) with other levels of models, where **PCNN+ATT** is a sentence-level model, **PCNN+ATT+SL** is an entity-pair level model, and **BGWA** is a word-level model. Among these different granularity models, AFPCNN has achieved significant improvements in recall metric. The best results are highlighted using bold fonts.

Table 1. AUC values, F1 scores, and P@N results of the proposed method and various baselines.

Models	Metrics (%))					
	AUC	F1 score	P@100	P@200	P@300	Mean
Mintz [14]	10.6	24.3	51.8	50.0	44.8	48.9
MultiR [8]	12.6	27.5	70.2	65.1	61.7	65.7
MIML [21]	12.0	25.3	70.9	62.8	60.9	64.9
PCNN [26]	32.5	39.2	72.3	69.7	64.1	68.7
PCNN+ATT [11]	34.8	42.3	76.2	73.1	67.4	72.2
BGWA [9]	36.0	43.1	75.2	74.1	71.4	73.6
PCNN+ATT+SL [12]	38.6	43.7	78.2	74.7	72.1	75.3
AFPCNN (Ours)	**40.3**	**45.1**	**84.2**	**78.1**	**76.4**	**79.6**

5.5 Discussion of different Attention Strategies

Different attention strategies have various formulas to compute attention scores. Our experiments compared these types on the AFPCNN model and found that the scaled dot product method is the least expensive and best-performing one, as shown in Table 2.

Table 2. AUC values, F1 scores, and P@N results of the proposed method and various baselines.

Attention mechanisms	AUC (%)	F1 score (%)	Time (min)
Additive Attention	38.3	44.2	220
Self-Attention	39.1	43.9	260
Scaled dot product	40.3	45.1	200

6 Conclusion and Future Works

In this paper, we proposed a novel attention-based feature combination method and adopted a sentence-level region feature for input representations, which produced a more reasonable sentence encoding for neural relation extraction models. Experiments have shown that our approach achieves significant improvements compared with the baseline models.

In future, we will work in the following aspects:

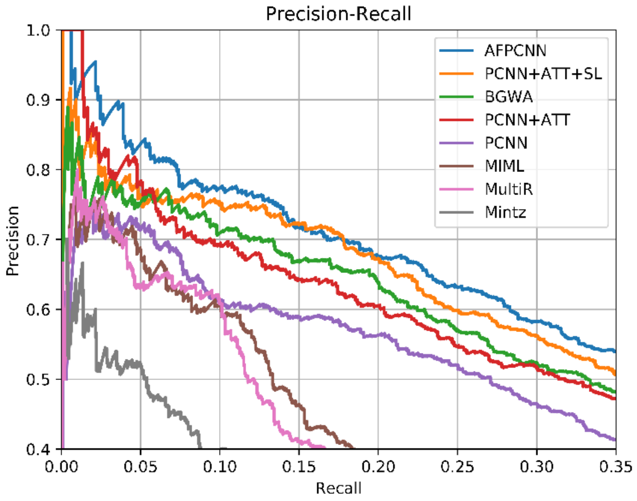


Fig. 4. Precision-recall curves for our models and various baseline models.

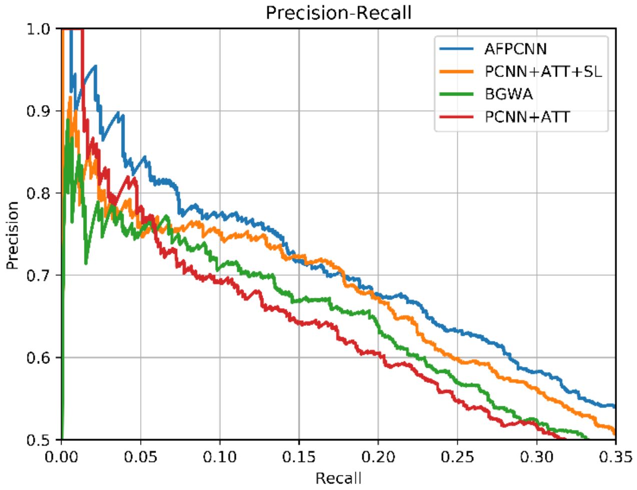


Fig. 5. Precision-recall curves for our feature-level model and other level models (PCNN+ATT+SL: entity-pair level, BGWA: word-level, PCNN+ATT: sentence-level).

(1) The proposed feature-level attention approach is extensible, and we will explore more features in the feature-level attention module and apply to other NLP tasks.

(2) The multi-instance learning framework is an effective way to reduce the noise for distant supervision. However, from the experimental results and previous work, the noise is far from being eliminated, so we will keep on the research of denoise methods for distant supervision.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Beltagy, I., Lo, K., Ammar, W.: Combining distant and direct supervision for neural relation extraction. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1858–1867 (2019)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
4. GuoDong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: Proceedings of the 43rd annual meeting on association for computational linguistics. pp. 427–434. Association for Computational Linguistics (2005)
5. Hashimoto, K., Miwa, M., Tsuruoka, Y., Chikayama, T.: Simple customization of recursive neural networks for semantic relation classification. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1372–1376 (2013)
6. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. pp. 94–99. Association for Computational Linguistics (2009)
7. Hinton, G.E., et al.: Learning distributed representations of concepts. In: Proceedings of the eighth annual conference of the cognitive science society. vol. 1, p. 12. Amherst, MA (1986)
8. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: 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. pp. 541–550. Association for Computational Linguistics (2011)
9. Jat, S., Khandelwal, S., Talukdar, P.: Improving distantly supervised relation extraction using word and entity based attention. arXiv preprint arXiv:1804.06987 (2018)
10. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. p. 22. Association for Computational Linguistics (2004)
11. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2124–2133 (2016)
12. Liu, T., Wang, K., Chang, B., Sui, Z.: A soft-label method for noise-tolerant distantly supervised relation extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1790–1795 (2017)
13. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)

14. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: 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. pp. 1003–1011. Association for Computational Linguistics (2009)
15. Nguyen, T.H., Grishman, R.: Relation extraction: Perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. pp. 39–48 (2015)
16. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
17. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 148–163. Springer (2010)
18. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
19. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. pp. 1201–1211. Association for Computational Linguistics (2012)
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
21. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: 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. pp. 455–465. Association for Computational Linguistics (2012)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
23. Wu, Y., Bamman, D., Russell, S.: Adversarial training for relation extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1778–1783 (2017)
24. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
25. Ye, Z.X., Ling, Z.H.: Distant supervision relation extraction with intra-bag and inter-bag attentions. arXiv preprint arXiv:1904.00143 (2019)
26. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1753–1762 (2015)
27. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., et al.: Relation classification via convolutional deep neural network (2014)
28. Zhang, D., Wang, D.: Relation classification via recurrent neural network. arXiv preprint arXiv:1508.01006 (2015)