Incorporating Relation Paths in Neural Relation Extraction

Wenyuan Zeng[1], **Yankai Lin**[2], **Zhiyuan Liu**[2]*, **Maosong Sun**[2]
[1]Department of Physics, Tsinghua University, Beijing, China
[2]State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract

Distantly supervised relation extraction has been widely used to find novel relational facts from plain text. To predict the relation between a pair of two target entities, existing methods solely rely on those direct sentences containing both entities. In fact, there are also many sentences containing only one of the target entities, which also provide rich useful information but not yet employed by relation extraction. To address this issue, we build inference chains between two target entities via intermediate entities, and propose a path-based neural relation extraction model to encode the relational semantics from both direct sentences and inference chains. Experimental results on real-world datasets show that, our model can make full use of those sentences containing only one target entity, and achieves significant and consistent improvements on relation extraction as compared with strong baselines. The source code of this paper can be obtained from `https://github.com/thunlp/PathNRE`.

1 Introduction

Knowledge Bases (KBs) provide effective structured information for real world facts and have been used as crucial resources for several natural language processing (NLP) applications such as Web search and question answering. Typical KBs such as Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007) and YAGO (Suchanek et al., 2007) usually describe knowledge as multi-relational data and represent them as triple facts. As the real-world facts are infinite and increasing every day, existing KBs are still far from complete. Recently, petabytes of natural-language text containing thousands of different structure types are readily available, which is an important resource for automatically finding unknown relational facts. Hence, relation extraction (RE), defined as the task of extracting structured information from plain text, has attracted much interest.

Most existing supervised RE systems usually suffer from the issue that lacks sufficient labelled relation-specific training data. Manual annotation is very time consuming and labor intensive. One promising approach to address this limitation is distant supervision. (Mintz et al., 2009) generates training data automatically by aligning a KB with plain text. They assume that if two target entities have a relation in KB, then all sentences that contain these two entities will express this relation and can be regarded as a positive training instance. Since neural models have been verified to be effective for classifying relations from plain text (Socher et al., 2012; Zeng et al., 2014; dos Santos et al., 2015), (Zeng et al., 2015; Lin et al., 2016) incorporate neural networks method with distant supervision relation extraction. Further, (Ye et al., 2016) considers finer-grained information, and achieves the state-of-the-art performance.

Although existing RE systems have achieved promising results with the help of distant supervision and neural models, they still suffer from a major drawback: the models only learn from those sentences contain both two target entities. However, those sentences containing only one of the entities could also provide useful information and help build inference chains. For example, if we know that "h is the father of e" and "e is the father of t", we can infer that h is the grandfather of t.

In this work, as illustrated in Fig. 1, we introduce a path-based neural relation extraction model

*Corresponding author: Z. Liu (liuzy@tsinghua.edu.cn).

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1768–1777
Copenhagen, Denmark, September 7–11, 2017.

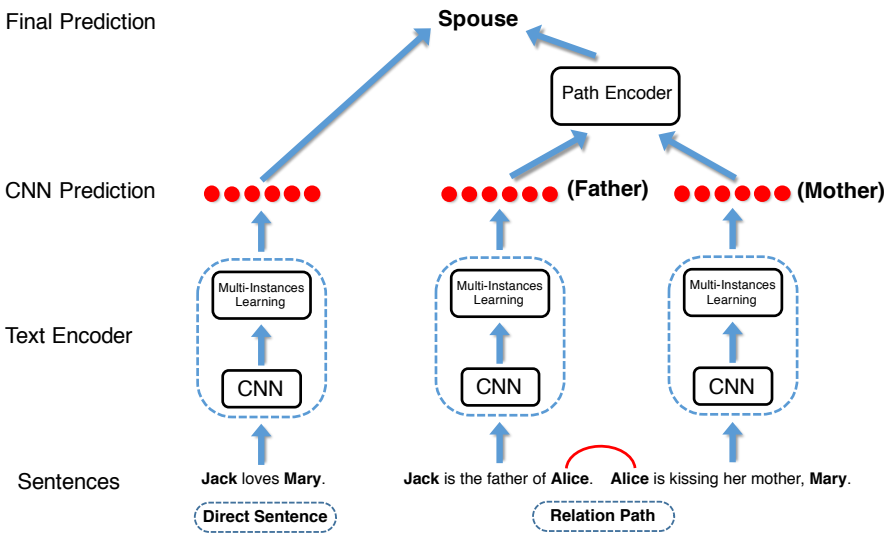

Figure 1: The architecture of our neural relation extraction model with relation paths.

with relation paths. First, we employ convolutional neural networks (CNN) to embed the semantics of sentences. Afterward, we build a relation path encoder, which measures the probability of relations given an inference chain in the text. Finally, we combine information from direct sentences and relation paths to predict the relation.

We evaluate our model on a real-world dataset for relation extraction. The experimental results show that our model achieves significant and consistent improvements as compared with baselines. Besides, with the help of those sentences containing one of the target entities, our model is more robust and performs well even when the number of noisy instances increases. To the best of our knowledge, this is the first effort to consider the information of relation path in plain text for neural relation extraction.

2 Related Work

2.1 Distant Supervision

Distant supervision for RE is originally proposed in (Craven et al., 1999). They focus on extracting binary relations between proteins using a protein KB as the source of distant supervision. Afterward, (Mintz et al., 2009) aligns plain text with Freebase, by using distant supervision . However, most of these methods heuristically transform distant supervision to traditional supervised learning, by regarding it as a single-instance single-label problem, while in reality, one instance could correspond with multiple labels in different scenarios and vice versa. To alleviate the issue, (Riedel et al., 2010) regards each sentence as a training instance and allows multiple instances to share the same label but disallows more than one label. Further, (Hoffmann et al., 2011; Surdeanu et al., 2012) adopt multi-instance multi-label learning in relation extraction. The main drawback of these methods is that they obtain most features directly from NLP tools with inevitable errors, and these errors will propagate to the relation extraction system and limit the performance.

2.2 Neural Relation Extraction

Recently, deep learning (Bengio, 2009) has been successfully applied in various areas, including computer vision, speech recognition and so on. Meanwhile, its effectiveness has also been verified in many NLP tasks such as sentiment analysis (dos Santos and Gatti, 2014), parsing (Socher et al., 2013), summarization (Rush et al., 2015) and machine translation (Sutskever et al., 2014). With the advances of deep learning, there are growing works that design neural networks for relation extraction. (Socher et al., 2012) uses a recursive neural network in relation extraction, and (Xu et al., 2015; Miwa and Bansal, 2016) further use LSTM. (Zeng et al., 2014; dos Santos et al., 2015) adopt CNN in this task, and (Zeng et al., 2015; Lin et al., 2016) combine attention-based multi-instance learning which shows promising results. However, these above models merely learn from those sentences which directly contain both two target entities. The important information of those relation paths hidden in the text is ignored. In this paper, we propose a novel path-based neural RE

model to address this issue. Besides, although we choose CNN to test the effectiveness of our model, other neural models could also be easily adapted to our architecture.

2.3 Relation Path Modeling

Relation paths have been taken into consideration on large-scale KBs for relation inference. Path Ranking algorithm (PRA) (Lao and Cohen, 2010) has been adopted for expert finding (Lao and Cohen, 2010), information retrieval (Lao et al., 2012), and further for relation classification based on KB structure (Lao et al., 2011; Gardner et al., 2013). (Neelakantan et al., 2015; Lin et al., 2015; Das et al., 2016; Wu et al., 2016) use recurrent neural networks (RNN) to represent relation paths based on all involved relations in KBs.(Guu et al., 2015) proposes an embedding-based compositional training method to connect the triple knowledge for KB completion. Different from the above work of modeling relation paths in KBs, our model aims to utilize relation paths in text corpus, and help to extract knowledge directly from plain text.

3 Our Method

Given a pair of target entities, a set of corresponding direct sentences $S = \{s_1, s_2, \cdots, s_n\}$ which contains this entity pair, and a set of relation paths $P = \{p_1, p_2, \cdots, p_m\}$, our model aims to measure the confidence of each relation for this entity pair. In this section, we will introduce our model in three parts: (1) **Text Encoder.** Given the sentence with two corresponding target entities, we use a CNN to embed the sentence into a semantic space, and measure the probability of each relation given this sentence. (2) **Relation Path Encoder.** Given a relation path between the target entities, we measure the probability of each relation r, conditioned on the relation path. (3) **Joint Model.** We integrate the information from both direct sentences and relation paths, then predict the confidence of each relation.

3.1 Text Encoder

As shown in Fig. 2, we use a CNN to extract information from text. Given a set of sentences of an entity pair, we first transform each sentence s into its distributed representation $\mathbf{s}$, and then predict relation using the most representative sentence via a multi-instance learning mechanism.

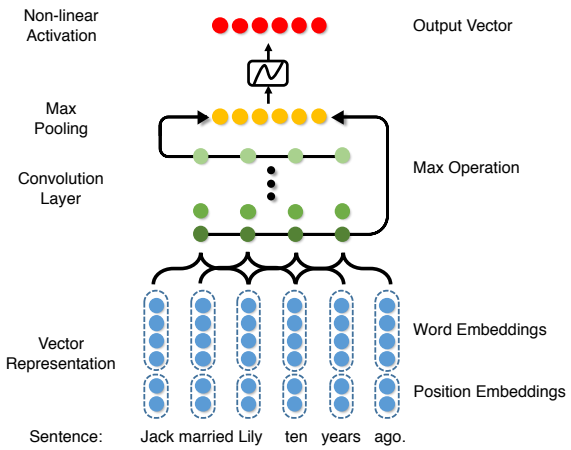

Figure 2: The architecture of CNN used for text encoder.

3.1.1 Input Vector

First, we transform the words $\{w_1, w_2, \cdots, w_l\}$ in sentence s into vectors of dimension d. For each word w_i, we use word embedding to encode its syntactic and semantic meanings, and use position embedding to encode its position information. We then concatenate both word embedding and position embedding to form the input vector of w_i for CNN. (See Figure 2.)

3.1.2 Convolution and Max-pooling Layers

When processing a sentence, it is a great challenge that important information could probably appear in all parts of that sentence. In addition, the length l of a sentence could also vary a lot. Therefore, we apply CNN to encode all local features regardless sentence length. We first apply a convolution layer to extract all possible local features, and then select the most important one via max-pooling layer.

To extract local features, the convolution layer first concatenates a sequence of word embeddings within a sliding window to be vector $\mathbf{q}_i$ of dimension $k \times d$:

$$\mathbf{q}_i = \mathbf{w}_{[i-k+1:i]} (1 \leq i \leq l + k - 1), \quad (1)$$

where k is the size of the window, and we also set all out-of-index words to be zero vectors. It then multiplies $\mathbf{q}_i$ by a convolution matrix $\mathbf{W} \in \mathbb{R}^{d_c \times (k \times d)}$, where d_c is the dimension of sentence embeddings. Hence, the output of convolution layer could be expressed as $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{l+k-1}\}$:

$$\mathbf{h}_i = \mathbf{W}\mathbf{q}_i + \mathbf{b}, \quad (2)$$

where $\mathbf{b}$ is a bias vector. Finally, the max-pooling layer takes a max operation, followed by a hyperbolic tangent activation, over the sequence of $\mathbf{h}_i$ to select the most important information, namely,

$$[\mathbf{s}]_j = \mathbf{tanh}(\max_i [\mathbf{h}_i]_j). \quad (3)$$

3.1.3 Multi-Instance Learning

Next, we apply a softmax classifier upon the sentence representation $\mathbf{s}$ to predict the corresponding relation. We define the condition probability of relation r as follows,

$$p(r|\theta, \mathbf{s}) = \frac{\exp(e_r)}{\sum_{i=1}^{n_r} \exp(e_i)}, \quad (4)$$

where e_i, a component of $\mathbf{e}$, measures how well this sentence matches relation r_i, and n_r is the number of relations. More specifically, $\mathbf{e}$ could be calculated from:

$$\mathbf{e} = \mathbf{U}\mathbf{s} + \mathbf{v}, \quad (5)$$

where $\mathbf{U} \in \mathbb{R}^{n_r \times d_c}$ is the coefficient matrix of relations and $\mathbf{v} \in \mathbb{R}^{n_r}$ is a bias vector.

We use multi-instance learning to alleviate the wrong-labeling issue in distant supervision, by choosing one sentence in the set of all direct sentences $S = \{s_1, s_2, \cdots, s_m\}$ which corresponds to the entity pair (h, t). Similar to (Zeng et al., 2015), we define the score function of this entity pair and its corresponding relation r as a max-one setting:

$$E(h, r, t|S) = \max_i p(r|\theta, \mathbf{s}_i). \quad (6)$$

where E reflects the direct information we derive from sentences. We can also set a random setting as a baseline:

$$E(h, r, t|S) = p(r|\theta, \mathbf{s}_i), \quad (7)$$

where s_i is randomly selected from S.

3.2 Relation Path Encoder

We use Relation Path Encoder to embed the inference information of relation paths. Relation Path Encoder measures the probability of each relation r given a relation path in the text. This will utilize the inference chain structure to help make predictions. More specifically, we define a path p_1 between (h, t) as $\{(h, e), (e, t)\}$, and the corresponding relations are r_A, r_B. Each of (h, e) and (e, t) corresponds to at least one sentence in the text. Our model calculates the probability of relation r conditioned on p_1 as follows,

$$p(r|r_A, r_B) = \frac{\exp(o_r)}{\sum_{i=1}^{n_r} \exp(o_i)}, \quad (8)$$

where o_i measures how well relation r matches with the relation path (r_A, r_B). Inspired by the work on relation path representation learning (Lin et al., 2015), our model first transforms relation r to its distributed representation, i.e. vector $\mathbf{r} \in \mathbb{R}^{d_R}$, and builds the path embeddings by composition of relation embeddings. Then, the similarity o_i is calculated as follows:

$$o_i = -\|\mathbf{r}_i - (\mathbf{r}_A + \mathbf{r}_B)\|_{L_1}. \quad (9)$$

Therefore, if $\mathbf{r}_i$ gets more similar to $(\mathbf{r}_A + \mathbf{r}_B)$, the conditioned predicting probability of r_i will become larger. Here, we make an implicit assumption that if r_i is semantically similar to relation path $p_i : h \xrightarrow{r_A} e \xrightarrow{r_B} t$, the embedding $\mathbf{r}_i$ will be closer to the relation path embedding $(\mathbf{r}_A + \mathbf{r}_B)$. Finally, for this relation path $p_i : h \xrightarrow{r_A} e \xrightarrow{r_B} t$, we define an relation-path score function,

$$G(h, r, t|p_i) = E(h, r_A, e)E(e, r_B, t)p(r|r_A, r_B), \quad (10)$$

where $E(h, r_A, e)$ and $E(e, r_B, t)$ measure the probabilities of relational facts (h, r_A, e) and (e, r_B, t) from text, and $p(r|r_A, r_B)$ measures the probability of relation r given relation path (r_A, r_B).

In reality, there are usually multiple relation paths between two entities. Hence, we define the inferring correlation between relation r and several sentence paths P as,

$$G(h, r, t|P) = \max_i G(h, r, t|p_i), \quad (11)$$

where we use max operation to filter out those noisy paths and select the most representative path.

3.3 Joint Model

Given any entity pair (h, t), those sentences S directly mentioning them and relation paths P between them, we define the global score function with respect to a candidate relation r as,

$$L(h, r, t) = E(h, r, t|S) + \alpha G(h, r, t|P), \quad (12)$$

where $E(h, r, t|S)$ models the correlation between r and (h, t) calculated from direct sentences,

$G(h, r, t|P)$ models the inferring correlation between relation r and several sentence paths P. α equals to $(1 - E(h, r, t|S))$ times a constant β. This term serves to depict the relative weight between direct sentences and relation paths, since we don't need to pay much attention on extra information when CNN has already given a confident prediction, namely $E(h, r, t|S)$ is large.

One of the advantages of this joint model is to alleviate the issue of error propagation. The uncertainty of information from Text Encoder and Relation Path encoder is characterized by its confidence, and could be integrated and corrected in this joint model step. Furthermore, since we treat relation paths in a probabilistic way, our model could fully utilize all relation paths, i.e. those always hold and those likely to hold.

3.4 Optimization and Implementation Details

The overall objective function is defined as:

$$J(\theta) = \sum_{(h,r,t)} \log(L(h, r, t)), \qquad (13)$$

where the summing runs over the log loss of all entity pairs in text and θ represents the model parameters. To solve this optimization problem, we use mini-batch stochastic gradient descent (SGD) to maximize our objective function. We initialize $\mathbf{W}_E$ with the results from Skip-gram model, and initialize other parameters randomly. We also adopt dropout (Srivastava et al., 2014) upon the output layer of CNN.

We implement our model using C++. We train our model on Intel(R) Xeon(R) CPU E5-2620, and the training roughly takes half a day. The word embedding and other parameters are updated via back-propagation simultaneously, while the relation path structure is extracted before training and stored afterward.

4 Dataset

We build a novel dataset for evaluating relation extraction task. We first describe the most commonly used previous dataset and then explain the reason and how we construct the new dataset.

4.1 Previous Datasets & Reasons for New Dataset

A commonly used benchmark dataset for this task was developed by (Riedel et al., 2010). This dataset was built by aligning Freebase (Dec. 2009 Snapshot) with New York Times corpus (NYT). There are 53 possible relationships between two entities, including a special relation type NA, meaning that there is no relation between head and tail entities. For each relational fact in a filtered Freebase dataset, a sentence from NYT would be regarded as a mention of this relation if both the head and tail entity appear in that sentence.

While this previous dataset has been frequently used for evaluating relation extraction systems, we observe some limitations of it. First, the relational facts are extracted from a 2009 snapshot of Freebase. Therefore, this dataset is too old to contain many updated facts. This will underestimate the performance of a relation extraction system, since some real-world facts are missing from the dataset and labeled as NA. Second, the relational facts in this dataset are scattered, i.e. there are not sufficient relation paths in this dataset, while relational facts in real-world always have connections with each other. Third, Freebase will no longer update after 2016.These limitations mean that this dataset is somewhat improper for evaluating RE systems.

Although other relation extraction datasets exist, e.g. ACE[1] and (Hendrickx et al., 2009), they are too small to train an effective neural relation extraction model. Moreover, each relational fact in (Hendrickx et al., 2009) only corresponds with one sentence, which prevents it from evaluating multi-instance relation extraction systems. Hence, we constructed a novel relation extraction dataset to address these issues, and will make it available to the community.

4.2 Dataset Construction

Datasets	Sets	# sentences	# entity pairs	# facts
Riedel et.al.	Train	522,611	281,270	18,252
	Valid	-	-	-
	Test	172,448	96,678	1,950
Ours	Train	647,827	266,118	50,031
	Valid	234,350	121,160	5,609
	Test	235,609	121,837	5,756

Table 1: Statistics of datasets.

Our dataset contains more updated facts and richer structures of relations, e.g. more relations / relation paths, as compared to existing similar datasets. The dataset is expected to be more similar to real-world cases, and thus be more appropriate for evaluating RE systems' performances.

We build the dataset by aligning Wikidata[2] re-

[1] https://catalog.ldc.upenn.edu/LDC2006T06

[2] https://www.wikidata.org/

lations with the New York Times Corpus (NYT). Wikidata is a large, growing knowledge base, which contains more than 80 million triple facts and 20 million entities. Different from Freebase, Wikidata is still in maintenance and could be easily accessed by APIs. We first pick those entities simultaneously appeared in both Wikidata and Freebase, and relational facts associated with them. Then, we filtered out a subset S, reserving those facts associating with the 99 highest frequency relations. This results in $4,574,665$ triples with $1,045,385$ entities and 99 relations.

Next, we align those facts with NYT corpus, following the assumption of distant supervision. For each pair of entities appearing in our S, we traverse the corpus and pick those sentences where both entities appear. These sentences will be regarded as mentions of this fact, and labeled by this relation type. To simulate noise in the real world, we also add sentences corresponding to "No Relation" entity pairs into our dataset. To get those "No Relation" instances, we first create a fake knowledge base S^- by randomly replacing the head or tail entities in triples, i.e., $S^- = \{(h', r, t)\} \cup \{(h, r, t')\}$ and then align them with NYT corpus. Finally, we randomly split all those selected sentences into training, validation and testing set, assuring that a relational fact could be only mentioned by sentences in one set. The statistics of our dataset and (Riedel et al., 2010) are listed in Table 1.

5 Experiments

Following the previous work (Mintz et al., 2009), we evaluate our model by extracting relational facts from the sentences in test set, and compare them with those in Wikidata. We report Precision/Recall curves, Precision@N (P@N) and F1 scores for comparison in our experiments.

5.1 Initialization and Parameter Settings

In this paper, we use the word2vec tool [3] to pre-train word embeddings on NYT corpus. We keep the words which appear more than 100 times in the corpus as vocabulary. We tune our model on the validation set, using grid search to determine the optimal parameters, which are shown in boldface. We select learning rate for SGD $\lambda \in \{0.1, \mathbf{0.01}, 0.001\}$, the sentence embedding size $d_c \in \{50, 60, \cdots, \mathbf{230}, \cdots, 300\}$, the window size $k \in \{1, 2, \mathbf{3}, 4, 5\}$, and the mini-batch size $B \in \{40, \mathbf{160}, 640\}$. Besides, we select the relation embeddings size $d_R \in \{5, 10, \cdots, \mathbf{40}, \cdots, 60\}$, and the weight for information from relation paths $\beta \in \{0.01, 0.1, 0.2, \mathbf{0.5}, 1, \cdots, 5\}$. For other parameters which have little effect on the system performance, we follow the settings used in (Zeng et al., 2015): word embedding size d_w is 50, position embedding size d_p is 5 and dropout rate p is 0.5. For training, the iteration number over all training data is 25.

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

5.2 Effectiveness of Incorporating Relation Paths

5.2.1 Precision-Recall Curve Comparison

To demonstrate the effect of our approach, we empirically compare it with other neural relation extraction methods via held-out evaluation. (1) **CNN+rand** represents the CNN model reported in (Zeng et al., 2014). (2) **CNN+max** represents the CNN model with multi-instances learning used in (Zeng et al., 2015). (3) **Path+rand/max** is our model with those two multi-instance settings. We implement (1), (2) by ourselves which achieve comparable results as reported in those papers.

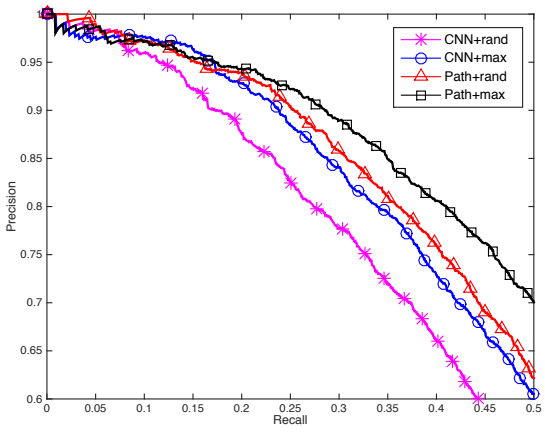

Figure 3: Aggregate precision/recall curve for CNN+rand, CNN+max, Path+rand, Path+max.

Fig. 3 shows the precision/recall curves of all methods. From the figure, we can observe that: (1) Our methods outperform their counterpart methods, achieving higher precision over almost entire range of recall. They also enhance recall by 20% without decrease of precision. These results prove the effectiveness of our approach. We notice that the improvements of our methods over

Test Settings (Noise)	75%				85%				95%			
P@N (%)	10%	20%	50%	F1	10%	20%	50%	F1	10%	20%	50%	F1
CNN+rand	86.7	67.0	38.9	57.5	84.6	66.4	37.5	55.0	79.9	61.8	35.2	51.4
CNN+max	86.0	68.5	38.3	57.2	85.4	67.6	37.7	56.5	84.4	66.0	36.6	54.8
Path+rand	**89.4**	**71.7**	**39.9**	59.3	88.2	70.2	39.0	58.1	86.0	67.2	37.0	55.6
Path+max	89.0	71.5	39.8	**59.6**	**89.0**	**71.4**	**39.6**	**59.4**	**88.6**	**71.0**	**39.1**	**59.1**

Table 2: P@N and F1 for relation extraction in texts containing different percentage of no-relation facts.

baselines are relatively small at small recall value, which corresponds to high predicting confidence. This phenomenon is intuitive since our joint model could dynamically leverage the importance of direct sentence and relation paths, and tends to trust the Text Encoder when the confidence is high. (2) As the recall increases, our models exhibit larger improvements compared with CNN in terms of percentage. This is due to the fact that sometimes CNNs cannot extract reliable information from direct sentences, while our methods could alleviate this issue by considering more information from inference chains, and thus still maintain high precision. (3) Both CNN+max and Path+rand are variations of CNN+rand, aiming to alleviate the problem of noisy data. We see that Path+rand outperforms CNN+max over all range, which indicates that considering path information is a better way to solve this issue. Meanwhile, combining paths information and max operation, Path+max, gives the best performance. (4) Path+rand shows a larger improvement over CNN+rand, compared with those of Path+max and CNN+max. This furthermore proves the effectiveness of considering relation path information: CNN+rand has much more severe problem suffering from noise, so using our method to incorporate paths information to alleviate this issue could perform better.

5.2.2 Comparison on Long Tail Situation

	$N_s \leq 1$	$N_s \leq 2$	$N_s \leq 5$	All
CNN+rand	53.9	54.0	52.0	51.4
Path+rand	**58.4**	**58.1**	**56.0**	**55.6**
CNN+max	57.8	58.3	56.5	55.7
Path+max	**63.6 (+5.8)**	**62.5 (+4.2)**	**60.2 (+3.7)**	**59.1 (+3.4)**

Table 3: F1 score for long tail situation.

Real-world data follows long-tail distribution (power law). In the testing set, we also observe a fact that about 40% triple facts appear only in single sentence, and thus a multi-instance relation extraction system, e.g. CNN+max, could only rely on limited information and the multi-instance mechanism will not work well. Our system, on the contrary, can still utilize information from relation paths in this case, and is expected to perform much better in the long tail situation.

We evaluate the models on different parts of the long-tail distribution. To get testing instances from different parts of the distribution, we extract all the triple facts appearing less than N_s sentences in the testing set, and those sentences associating with them. All text related to 'No Relation' entity pair are also reserved in order to simulate noise. We then evaluate different models on those sampled testing set, and report the results of F1 score in Table 3.

From Table 3, we could observe that: (1) Incorporating relation paths is indeed effective in predicting relations, and our models have significant improvements compared with the baselines. (2) Path+max indeed has larger improvements over CNN+max when N_s is small, which is consistent with our previous expectation. Also notice that the gap between Path+rand and CNN+rand is relatively constant. This is due to the fact that both these methods only use one random sentence, regardless of how many sentences there are associating with an entity pair.

5.3 Model Robustness under Different Percentages of Noise

In the task of relation extraction, there are lots of noise in text which may hurt the model's performance. More specifically, "No Relation" entity pair is a kind of noise, since "No Relation" could actually contain many unknown relation types, and thus might confuse the relation extraction systems. Therefore, it is important to verify the robustness of our model in the presence of massive noise. Here, we evaluate those models in three settings, with the same relational facts and different percentages of "No Relation" sentences in the testing sets. In each experiment, we extract top 20,000 predicting relational facts according to the model's predicting scores, and report the precision @top 10%, @top 20%, @top 50% and F1 score in Table 2.

From the table, we can see that: (1) In terms of all evaluations, our models achieve the best perfor-

	Relation	Text
Path #1	mother	**Rebecca** gave birth to twin sons, **Esau** and Jacob, ...
Path #2	has_child	...**Isaac**'s marriage to Rebecca, by whom he has two sons, **Esau** and jacob, ...
Test	spouse	... **Isaac** and **Rebecca** and the female and male evil spirits ...
Path #1	shares_border_with	... in **Somalia**, ... soldiers and marines stationed in neighboring **Djibouti** ...
Path #2	shares_border_with	... **Ethiopia** have had the effect of making neighboring **Djibouti** ...
Test	shares_border_with	The next day, **Ethiopia** struck, its military pushing deep into **Somalia** ...

Table 4: Some representative examples of inference chians in NYT corpus. The bold is target entities.

mance as compared with other methods in all test settings. It demonstrates the effectiveness of our approach. (2) Even though the scores of all models drop as the increasing of noise, we find that Path+rand/max's scores decrease much less than their counterparts. This result proves the effectiveness of taking inference chains into consideration. Since we utilize more information to make predictions, our model is more robust to the presence of mass noise.

5.4 Effectiveness of Learned Features in Zero-Shot Scenario

It has been proved that CNN could automatically extract useful features, encoding syntactic and semantic meaning of sentences. These features are sometimes fed to subsequent models to solve other tasks. In this experiment, we demonstrate the effectiveness of the extracted features from our model. Since CNN-based models have already succeeded in extracting relations from single sentences, we set our experiment in a new scenario: predicting the relation between entities which have not appeared in the same sentence.

A natural approach is to build a relation path between this zero-shot entity pair. We assume that we can make a prediction about (h, t), once we know the information of (h, e) and (e, t). Therefore, we build the training set by extracting all such relation paths and their sentences from training text, and similar for testing set. To test the effectiveness of features, we encode sentences by CNN+rand/max, Path+rand/max respectively, and then feed the concatenation of sentence vectors to a logistic classifier.

Feature	Accuracy
CNN+rand	56.9
CNN+max	57.3
Path+rand	58.5
Path+max	**60.4**

Table 5: Accuracy of different models in zero-shot situation.

From Table 5, we could observe that: (1) The result using CNN+rand features is comparable to the result using CNN+max features. It shows that using max operation to train the features does not greatly improve the features' behavior in this task, even though it performs well in previous tasks. The reason is that, both CNN+rand and CNN+max only encode the information from a single sentence, and they are unable to capture the correlations between relations. (2) Feature from Path+rand/max shows its effectiveness over those from other methods. It indicates that our method is able to model the correlations between relations, while also keeps the syntactic and semantic meaning of a sentence. Therefore, the features extracted from Path+rand/max are useful for a wider range of applications, especially in those tasks which need the information from relations.

5.5 Case study

Table 4 shows some representative inference chains from the testing dataset. These examples can not be predicted correctly by the original CNN model, but are later corrected using our model. We show the test instances and their correct relations, as well as the inference chains the model uses. In the first example, the test sentence does not directly express the relation `spouse`, the proof of this relation appears in a further context in NYT. However, using path#1 and path#2, we could easily infer that *Rebecca* and *Issac* are `spouse`. The second example doesn't show the relation either. But with the help of intermediate entity, *Dijibouti*, our model predicts that *Somalia* shares the border with *Ethiopia*. Note that this inference chain doesn't always hold, but our model could capture this uncertainty well via a softmax operation. In general, our model can utilize common sense from inference chains. It helps make correct predictions even if the inference is not explicit.

6 Conclusion and Future Work

In this paper, we propose a neural relation extraction model which encodes the information of

relation paths. As compared to existing neural relation extraction models, our model is able to utilize the sentences which contain both two target entities and only one target entity and is more robust for noisy data. Experimental results on real-world datasets show that our model achieves significant and consistent improvements on relation extraction as compared with baselines.

In the future, we will explore the following directions: **(1)** We will explore the combination of relation paths from both plain texts and KBs for relation extraction. **(2)** We may take advantages of probabilistic graphical model or recurrent neural network to encode more complicated correlations between relation paths, e.g. multi-step relation paths, for relation extraction.

Acknowledgments

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61572273, 61661146007), China Association for Science and Technology (2016QNRC001), and Tsinghua University Initiative Scientific Research Program (20151080406).

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. *Dbpedia: A nucleus for a web of open data*. Springer.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of KDD*, pages 1247–1250.

Mark Craven, Johan Kumlien, et al. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of ISMB*, volume 1999, pages 77–86.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2016. Chains of reasoning over entities, relations, and text using recurrent neural networks. *arXiv preprint arXiv:1607.01426*.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom M Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of EMNLP*, pages 833–838.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of EMNLP*, pages 318–327.

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

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL-HLT*, pages 541–550.

Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of EMNLP*, pages 529–539.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of EMNLP-CoNLL*, pages 1017–1026.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2015. Modeling relation paths for representation learning of knowledge bases. *Proceedings of EMNLP*.

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

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of ACL*, pages 1105–1116.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base inference. In *2015 AAAI Spring Symposium Series*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *Proceedings of EMNLP*.

Cıcero Nogueira dos Santos and Maıra Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING*.

Cıcero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL*, volume 1, pages 626–634.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of ACL*. Citeseer.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP-CoNLL*, pages 1201–1211.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of WWW*, pages 697–706. ACM.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, pages 455–465.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112.

Jiawei Wu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Knowledge representation via joint learning of sequential text and knowledge graphs. *arXiv preprint arXiv:1609.07075*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of EMNLP*, pages 1785–1794.

Hai Ye, Wenhan Chao, and Zhunchen Luo. 2016. Jointly extracting relations with class ties via effective deep ranking. *arXiv preprint arXiv:1612.07602*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*.

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