A Relation Extraction Method Based on Entity Type Embedding and Recurrent Piecewise Residual Networks

Yuming Wang[1,3(✉)], Huiqiang Zhao[1], Lai Tu[1], Jingpei Dan[2], and Ling Liu[3]

[1] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China
ymwang@mail.hust.edu.cn
[2] College of Computer Science, Chongqing University, Chongqing 400044, China
[3] School of Computer Science, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract. Relation extraction is an important while challenging task in information extraction. We find that existing solutions can hardly extract correct relation when the sentence is long and complex or the firsthand trigger word does not show. Inspired by the idea of fusing more and deeper information, we present a new relation extraction method that involves the types of entities in the joint embedding, namely, Entity Type Embedding (ETE). An architecture of Recurrent Piecewise Residual Networks (RPRN) is also proposed to cooperate with the joint embedding so that the relation extractor acquires the latent representation underlying the context of a sentence. We validate our method by experiments on public data set of New York Times. Experiment results show that our method outperforms the state-of-the-art models.

Keywords: Relation extraction · Entity Type Embedding · Recurrent neural networks · Residual Networks

1 Introduction

Information extraction aims at extracting structured information from large-scale text corpora, of which the main task is to recognize entities, the relation of the entity pair and the events involved [1]. The extracted facts of the relations can support a wide range of applications like Semantic Search, QA and etc [2]. To extract relations, most of the current solutions employ supervised training methods to learn from labelled corpus. These solutions usually use word embedding [3,4] and position embedding [5] in the representation layer. The embedded vectors then pass through a neural network and they are jointly trained with the labelled data. However, based on our experiment on the New York Times corpus data, existing attention-based solutions fail to extract relation in the following challenging cases as illustrated in Fig. 1.

K. Chen et al. (Eds.): BigData 2019, LNCS 11514, pp. 33–48, 2019.
https://doi.org/10.1007/978-3-030-23551-2_3

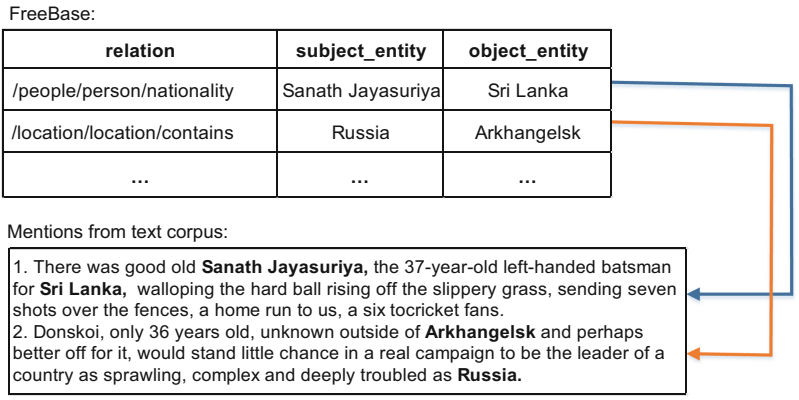


Fig. 1. An example of relation extraction model.

1. Cases that sentences have no firsthand trigger word [6] of the corresponding relation, such as the first example in Fig. 1.
2. Cases that the sentences are long and the position of the subject entity is far apart from the object entity, such as the second example in Fig. 1.

We observe that the extraction failure is due to the missing of latent information hidden deep in the context. To address this issue, we propose a method based on joint embedding and modified residual network. First we involve extra knowledge in the representation layer. More specifically, we introduce an Entity Type Embedding (ETE) module that maps the types of subject and object entities into vectors. The vectors then pass through a CNN network and are added to the encoded vector of word and position embedding with trained weights. Secondly, we design an new encoder module based on Recurrent Piecewise Residual Networks (RPRN). The architecture of RPRN increases the depth of the neural network so that the encoder is able to extract latent information from the sentence.

We evaluate our method on the New York Times dataset both qualitatively and quantitatively. Results show that our method can correctly extract the relations in the challenging cases that existing attention-based solutions fail. And the proposed solution also provides a better overall performance over all baseline methods.

In the following sections, we begin with a brief review of the related work in Sect. 2. And then, in Sect. 3, we represent our approach for relation extraction and elaborate the core procedure of our algorithm. Section 4 provides quantitative and qualitative experimental results. Finally, we conclude our work in Sect. 5.

2 Related Work

For relation extraction, current supervised method may extract more features and obtain higher precision and recall. However, the main problem of such method is time consuming and requiring intensive labor of labeling corpus manually. In response to the limitation above, Mintz et al. [7] applied Distant Supervision (DS) to relation extraction, they aligned the New York Times (NYT) corpus with the large-scale knowledge base of Freebase. DS assumes that the entity pair mentioned in a sentence implies the semantic relation of it in knowledge base. Riedel et al. [8] relaxed the assumption of DS, of which error rate reduced by 31% compared to Mintz. To break through the limitation that DS assumes each entity pair corresponds to only one relation, Hoffmann et al. [9] proposed a novel model of Multi-Instance Multi-Label to proceed relation extraction, which allows for the scenario existing multiple relations between an entity pair. As for the serious flaw of generating negative examples, Bonan et al. [10] proposed an algorithm that learns from only positive and unlabeled labels at the pair-of-entity level.

Although having achieved preferable results, the methods based on traditional machine learning rely on preprocessing such as Part-of-Speech tagging (POS) [11], Semantic Role Labeling (SRL) [12] and so on. Errors may exist during the preprocessing and may be propagated and amplified in relation extraction.

It is of much concern that deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [13]. Zeng et al. [14] first adopted Convolutional Neural Networks (CNN) while modeling relation extraction. To tackle the time consuming and manual labeling issues, the authors further integrated a new CNN with DS for relation extraction in [15].

Miwa et al. [16] proposed an end-to-end neural network, which utilized bidirectional Long Short Term Memory (LSTM) [17] network to model relation extraction. Peng et al. [18] and Song et al. [19] explored a general relation extraction framework based on graph LSTM that can be easily extended to cross-sentence n-ary relation extraction. To alleviate performance degradation caused by noise instance, Lin et al. [20] applied selective attention mechanism to relation extraction, which pledges that the weight of effective instance will rise and the noise one will decline. In addition, Wang et al. [21] and Du et al. [22] proposed a novel multi-level attention mechanism for relation extraction. Wu et al. [23] and Qin et al. [24] introduced the process of adversarial training to relation extraction to further enhance its generalization capacity.

3 Methodology

In this section, we represent the details of our approach, and elaborate the core procedure of our algorithm in the form of pseudo code.

3.1 Overview

The overall architecture of our model is sketched in Fig. 2, which is mainly divided into the following parts:

1. Embedding layer: Word embedding, position embedding and Entity Type Embedding (ETE) work jointly as the distributed representation of the model;
2. Encoder Layer: Word and position embedding output is encoded by a RPRN based encoder and ETE output is encoded with a CNN based encoder;
3. Attention Layer: To relieve the performance degradation of noisy instances, we employ a selective attention mechanism to pick out the positive instances while training.

Take the instance in Fig. 2 as an example, the top table represents the word embedding and position embedding, where word embedding is pre-trained and position embedding is initialized randomly, and the bottom table represents the ETE. The embedding of concatenating word with position goes through the architecture of RPRN, and then concatenates with the output of encoding ETE. After encoding, the instances possessing the same entity pair will be put into the same bag, and then the attention based selector is going to pick out positive instances from the bag as far as possible for the training relation. Finally, we will get a vector representing the latent pattern of the training relation, on which the last layer of classifier bases to output the most probable relation.

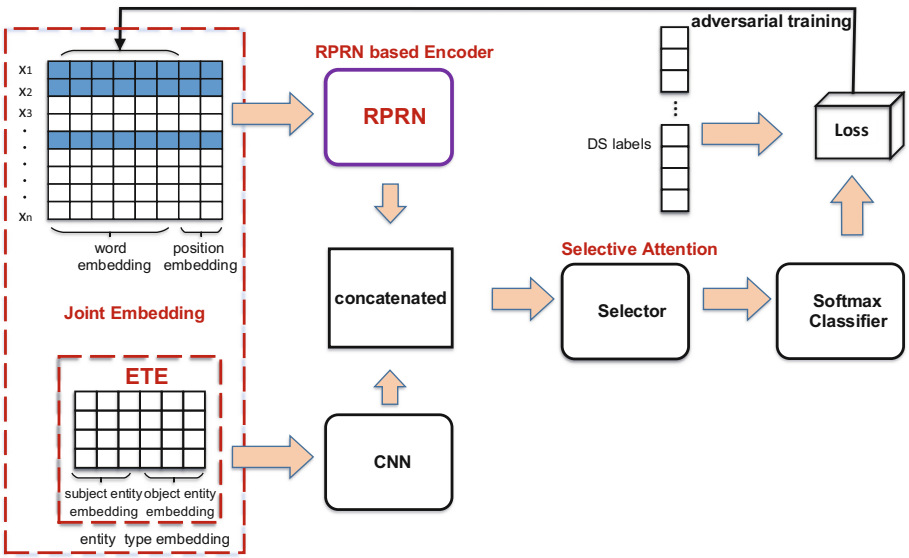


Fig. 2. The overall architecture of our model for relation extraction.

3.2 Joint Embedding

The distributed representation layer of relation extraction composes of word embedding, position embedding and ETE, whether it represents adequately makes a difference directly on the effects of the model.

(1) Word Embedding. Word embedding is a distributed representation of words to be mapped from a high-dimensional word space to a low-dimensional vector space, where the semantically similar words are close to each other as well. It can be seen that word embedding can express the semantic relation among the words. In our model, we utilize skip-gram [4] to pre-train word vector, but the pre-trained word vector is merely served as the initialization of the word embedding, afterwards it will be updated while training.

(2) Position Embedding. In order to exploit the information of position of each word in the sentence, position embedding [5] is applied to relation extraction. The information of position aforementioned denotes the relative distances (illustrated in Fig. 3) between each word in the sentence and the corresponding entity pair, and then map the relative distances of each word to a vector space. Besides, each word in a sentence correspond to two vectors initialized randomly, then the two vectors will be updated constantly along with continually training.

Fig. 3. The relative position between word and the corresponding entity.

(3) Entity Type Embedding. As can be seen from Fig. 1, the information of entity type plays an important role in relation extraction. Yet the structure of entity type is widely divergent from word embedding and position embedding, therefore it can not be served as the distributed representation layer that simply concatenating ETE with word embedding and position embedding. The scheme we adopted is that the ETE to be served as a part of representation is encoded separately with CNN, of which process is sketched in Fig. 2.

In our model, the dimension of word embedding is $d_w = 50$, the position embedding's is $d_p = 5$, and then joint together to transform an instance of training into a matrix $S \in \mathbb{R}^{n \times d}$, n = 120 denotes maximum length of sentences in the corpus, $d = d_w + 2 \times d_p$ denotes the dimension of representation layer after jointing word embedding and position embedding. The matrix S will be fed to the encoder of RPRN. The dimension of the representation for ETE is $d_{et} = 12$, the ETE represent as $E_{et} \in \mathbb{R}^{2 \times n_{et} \times d_{et}}$, where $n_{et} = 100$ denotes the maximum number of types for an entity over all corpus.

3.3 RPRN Based Encoder

Since ETE is introduced to the embedding layer and is encoded separately, the encoder in this model composes of two parts, a RPRN based encoder for

word and position embedding, and a CNN based encoder for ETE. Besides, the architecture of RPRN is illustrated in Fig. 4.

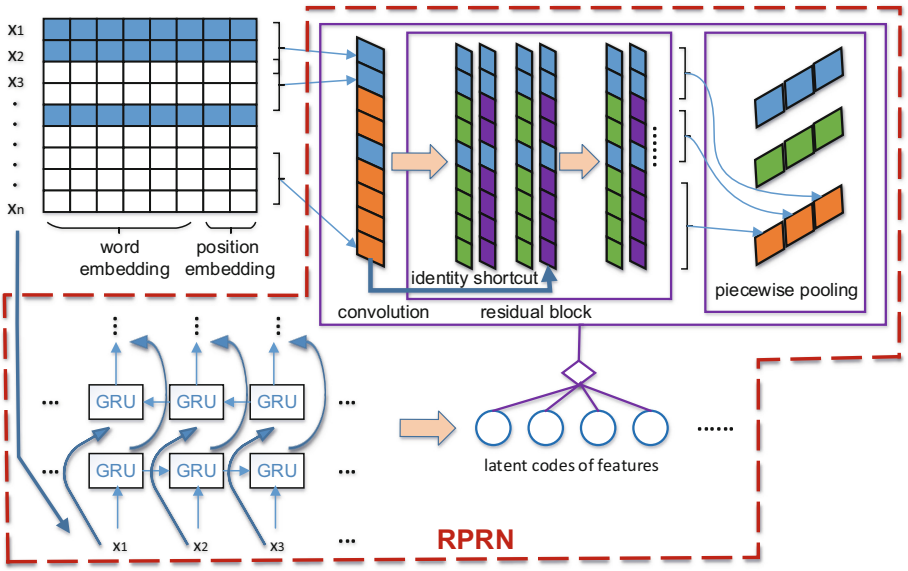


Fig. 4. The principle of the architecture of RPRN.

(1) Sequence Learning. We employ bidirectional RNN to learn the context information of the sentence, and we use Gated Recurrent Unit (GRU) [25] as the recurrent unit. GRU can deal with the problem of long-term dependency, and is more streamlined than LSTM. In our model, the gated unit is simplified into two gates, namely update gate z_t and reset gate r_t, which are derived as Eqs. (1) and (2) respectively.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

The output status of GRU is h_t, derived as Eq. (4), which is determined by the updated and reserved information based on the last time's:

$$h_t^{'} = g(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h_t^{'} \tag{4}$$

(2) Residual Networks. To learn finer features, deeper neural network is preferred. However, deeper neural network has to face the challenge of gradient vanishing or exploding [26], which makes the model hard to optimize. Therefore, we use a residual network [27] based model in the encoder as illustrated in Fig. 2.

We modify the identity mapping part and pooling part where the size of identity block is $I_s = 2$, and the number of blocks is $I_n = 3$. The RPRN with CNN based encoder for ETE altogether comprise of 12 layers in this model. The residual function is represented as Eq. (5), where x is the input and H is the output function of identity block.

$$F(x) = H(x) - x \tag{5}$$

(3) Piecewise Pooling. For extracting the semantic relation between the entity pair, we employ piecewise pooling to learn structured information, which is an effective approach of structured learning [28]. We introduce mask information represented as $M \in \mathbb{R}^{m \times n}$ and its embedding represented as M_e = [[0, 0, 0], [1, 0, 0], [0, 1, 0], [0, 0, 1]] to implement the piecewise pooling, where m denotes the batch size. The result after the operating of piecewise pooling is $P \in \mathbb{R}^{m \times (3*h)}$ derived as Eq. (6), where h denotes the hidden size.

$$P = \phi(\pi(\epsilon(\gamma(M_e, M) \times c, 2) + \epsilon(C, 3)) - c, [-1, 3 * h]) \tag{6}$$

Regarding the parameters involved in the equation above, ϕ denotes the function to adjust the shape of tensor, π is to proceed piecewise max pooling, ϵ is to expand a certain dimension of the tensor, γ denotes the function to look up embedding of mask, c is a constant, and C is a output tensor of the convolutional part. The principle of piecewise pooling is illustrated in Fig. 5.

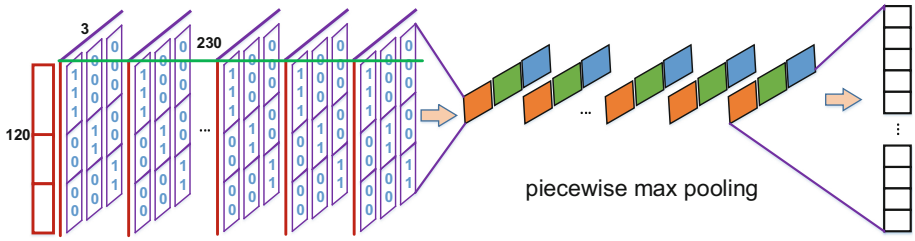


Fig. 5. The principle of piecewise pooling.

As a part of RPRN, the layer of piecewise pooling is designed to obtain principal features from the output of previous layer. Piecewise pooling, in the context of this paper, means dividing three parts according to the positions of entity pair in a sentence, and then each part proceed max pooling respectively.

The max pooling is utilized for ETE, and its hidden size is $h_{et} = 80$. The result of the operation for max pooling is then concatenated with the piecewise pooling's to be served as the encoder of relation extractor, and its output will be fed to the instance selector.

3.4 Selective Attention

The method of DS is usually applied to relation extraction to solve the problem of time consuming and labor intensive for labeling corpus manually, yet it simultaneously confronts with error labeling. The initial assumption of DS serves all the instances in a bag as positive examples, accordingly there are plenty of noise instances involving in training. For this issue, Riedel et al. [8] adopted serving the most likely one as positive example for training, which tremendously alleviates the performance reduction caused by noise instances. Meanwhile, such method discard so many effective instances and then hinder improving the model of relation extraction. Thus Lin et al. [20] applied the selective attention based mechanisms to relation extraction, and its principle is sketched in Fig. 2, the output vector A is derived as Eq. (7).

$$A = \omega\left(\mu\left\{\epsilon\left(\frac{exp^{H\cdot\gamma(R,y)+bias}}{\sum_{k=1}^{b} exp^{H_k\cdot\gamma(R_k,y_k)+bias_k}}, 0\right), H\right\}\right) \tag{7}$$

Algorithm 1. The core procedure of our algorithm

Input: training corpus $B = \{B_1, B_2, \cdots, B_m\}$, hyper-parameters K, pre-trained word vector V^w.

initialize parameters Θ, V^p, and V^{et}
$B^w, B^p, B^{et} \leftarrow \gamma(V^w, B^w), \gamma(V^p, B^p), \gamma(V^{et}, B^{et})$
$B^{wp} \leftarrow concat(B^w, B^p, -1)$
for l to K^l do
  $B \leftarrow shuffle(B)$
  $T \leftarrow ceil(length(B)/K^t)$
  for t to T do
    $H_{t-1}^{wp} \leftarrow$ feed Eq. (1) to Eq. 4 on B_t^{wp}
    $H_{t-2}^{wp} \leftarrow$ feed Eq. (5) on B_t^{wp}
    $H_{t-2}^{wp} \leftarrow$ feed Eq. (6) on H_{t-2}^{wp}
    $H_t^{wp} \leftarrow concat(H_{t-1}^{wp}, H_{t-2}^{wp}, -1)$
    $H_t^{et} \leftarrow cnn(B_t^{et})$
    $H_t \leftarrow concat(H_t^{wp}, H_t^{et}, -1)$
    foreach $H_{t-i} \in H_t$ do
      pick out positive instance as far as possible
      apply Eq. (7) to H_{t-i}
    end
    $O_t \leftarrow softmax(H_t)$
    then update the parameters Θ
    $\Theta \leftarrow \Theta + \nabla L_\Theta(O_t)$
  end
end

Output: extracted relation B^r

In the Eq. (7) above, ω is a squeeze function to strip all the dimensions valued 1 for the tensor, μ denotes matrix multiplication, H denotes the output of encoder, R denotes relation matrix, y is the ground-truth of corpus, $bias$ denotes a bias vector, b denotes the bag size, and other notations have been explained above.

Adopting the attention based mechanisms described above, the sentences that indeed express their relations involved in knowledge base can be picked out as positive instances, and those that don't will be served as noises. More specifically, such selective attention based mechanisms pledges that the weights of effective instances will rise and the noise one will decline, which effectively alleviates the performance degradation caused by noise instances.

For the last layer, we adopt softmax to classify the relation and serve cross-entropy as the loss function. Besides, we introduce the adversarial process [23] to perturb the input representation while training so that the model acquires better generalization capacity on the testing set.

In summary, the core procedure for implementing our model is elaborated in Algorithm 1. As for the notations in Algorithm 1, we will make some notes necessarily in the next moment. B, as the preprocessed training corpus, mainly contains the information of word, position and entity type, that's B^w, B^p, B^{et}. Θ denotes the parameters overall the training process of the model. V^p denotes position vector, V^{et} denotes entity type vector, which are all initialized randomly and subsequently optimized progressively with continuing training. the hyper-parameters K directly utilized in Algorithm. 1 are max epoch K^l and batch size K^t. Besides, H denotes the output of hidden layer, O denotes the final output of last layer, and L_Θ is the loss function of the entire model.

4 Experiments

In this section, we present the comparative results of our relation extraction method and several baseline solutions. We also compare the performance of different combinations of proposed sub-modules.

4.1 Experimental Setup

We conduct an experiment on the public dataset of New York Times (NYT) in this work, which is extensively used in relation extraction. The training set of NYT contains 522,611 sentences, 281,270 entity pairs, and 18,252 relation facts; The testing set contains 172,448 sentences, 96,678 entity pairs, and 1,950 relation facts. The entity pair, represented as <subject_guid#object_guid>, will obtain the score corresponding to each relation, and then it will subordinate to the relation obtaining the highest score.

(1) Hyper-parameters Settings. The best hyper-parameters, set as Table 1, are specified through cross validation and grid search in this work.

(2) Measurement Metrics. According to the combination of the prediction results and ground truth, the samples of corpus can be divided into four parts:

Table 1. Hyper-parameters settings.

Param name	Value	Description
word_dim	50	The dimension of word embedding
pos_dim	5	The dimension of pos embedding
et_dim	12	The dimension of ETE
hidden_size	230	The size of hidden layer for encoding word and position embedding
et_hidden_size	80	The size of hidden layer for encoding the information of ETE
ib_num	3	Num of identity block for RPRN
learning_rate	0.5	The learning rate for optimizer
drop_out	0.5	The drop out rate of discarding some neurons while training

True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN). We use precision, recall, F1 and AUC as the metrics which are widely used in relation extraction evaluation.

4.2 Baselines

We adopt four state-of-the-art solutions as baselines:

- **pcnn_one** (Zeng et al. [15]) served the most likely one in a bag as positive instance for training.
- **pcnn_att** (Lin et al. [20]) applied the selective attention based mechanisms to relation extraction, which effectively utilized plenty of positive instances in a bag.
- **bilstm_att** (Miwa et al. [16]) presented a novel end-to-end neural model by stacking bidirectional tree structured LSTM for relation extraction.
- **pcnn_att_ad** (Wu et al. [23], Qin et al. [24]) introduced adversarial training for relation extraction to further enhance its generalization capacity.

We also compare the performance of different combinations of proposed sub-modules, listed as follows:

- **ete_pcnn_att** utilizes ETE for the whole training data, compared to the previous model, still more proving the effectiveness of ETE.
- **ete_pcnn_att_ad** utilizes ETE and adversarial training jointly.
- **rprn_att** adopts the architecture of dubbed RPRN we devised.
- **rprn_att_ad** coalesces the architecture of RPRN and adversarial training.
- **ete_rprn_att** integrates the ETE with the architecture of RPRN we devised.
- **ete_rprn_att_ad** is a joint model based on ETE, the architecture of RPRN and adversarial training, which achieves best results in this paper.

4.3 Evaluation Results

Table 2 gives an overview of the performance of various models. Figures 6, 7, 8 and 9 further show the precision to recall performance of different method. Results show that all the metrics have been improved significantly. More specifically, the comparisons among various models reveal the following observations:

1. Fig. 6 shows that the performance of four baseline methods do not have distinct difference.
2. In Fig. 7a and b, we compare the models using ETE in two baselines respectively. We can see that ETE improves the baseline methods.
3. In Fig. 8a and b, RPRN based encoder is used alone. We find that single RPRN based encoder also improves the performance.

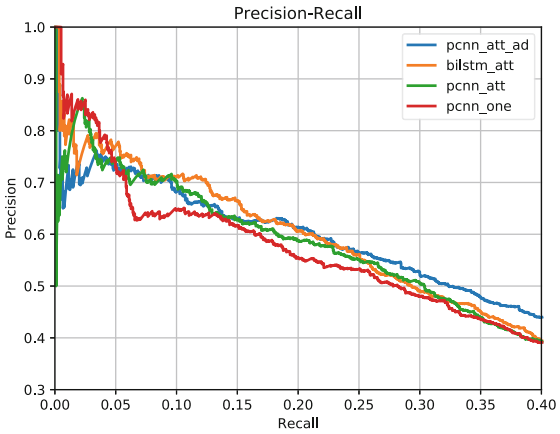
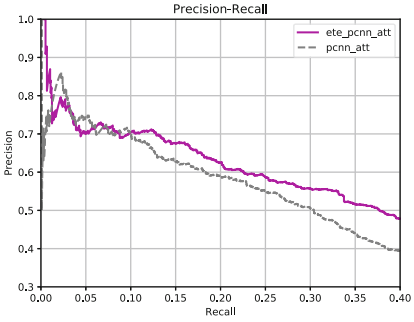
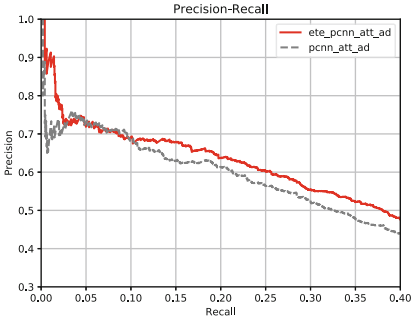


Fig. 6. Comparisons among the models of four **baselines**.

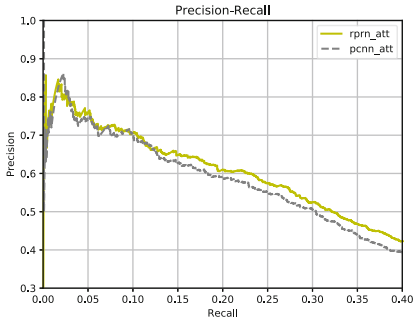


(a) Comparison between the models of **ete_pcnn_att** and **pcnn_att**.

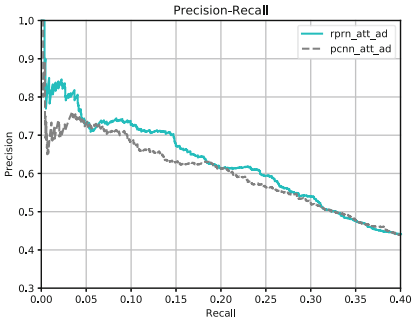


(b) Comparison between the models of **ete_pcnn_att_ad** and **pcnn_att_ad**.

Fig. 7. The curve of PR for proposed models utilizing the ETE.



(a) Comparison between the models of **rprn_att** and **pcnn_att**.



(b) Comparison between the models of **rprn_att_ad** and **pcnn_att_ad**.

Fig. 8. The curve of PR for proposed models adopting the architecture of RPRN.

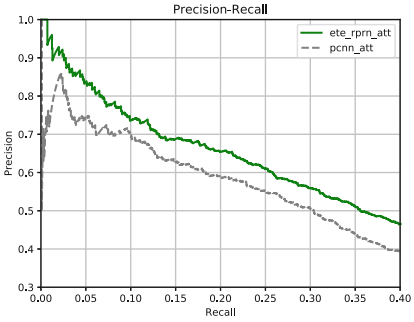
Table 2. Evaluation metrics for relation extraction.

Model name	F1-score	AUC-value
ete_rprn_att_ad	**0.4427**	**0.3965**
ete_rprn_att	**0.4409**	**0.3935**
ete_pcnn_att_ad	**0.4443**	**0.3898**
ete_pcnn_att	**0.4416**	**0.3882**
rprn_att_ad	**0.4198**	**0.3649**
rprn_att	**0.4135**	**0.3568**
pcnn_att_ad	0.4209	0.3518
bilstm_att	0.4020	0.3467
pcnn_att	0.3991	0.3414
pcnn_one	0.3984	0.3280

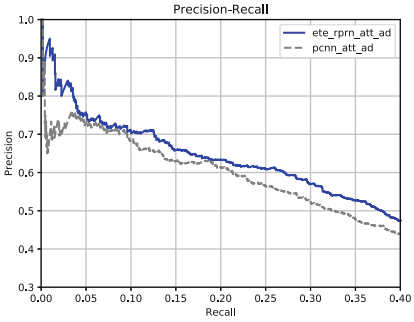
4. The models of (**ete_rprn_att_ad**) and (**ete_rprn_att_ad**), in Fig. 9a and b respectively, integrate the ETE with RPRN based encoder. Besides, the model of (**ete_rprn_att_ad**) achieves best results among all combinations and baseline methods.

4.4 Case Analysis

We further look at some specific cases with different models as shown in Table 3. The entity pair, namely 'Sanath Jayasuriya' and 'Sri Lanka', of the first sentence in Table 3 indeed express the 'nationality' relation, which is comparatively obscure and it is hard to be extracted purely through the information of the context in the sentence. The model (**pcnn_att**), quite understandably, can not



(a) Comparison between the models of **ete_rprn_att** and **pcnn_att**.



(b) Comparison between the models of **ete_rprn_att_ad** and **pcnn_att_ad**.

Fig. 9. The curve of PR for proposed models integrating ETE and RPRN.

Table 3. Cases for relation extraction.

Sentence	Extraction result	
	ete_pcnn_att	pcnn_att
There was good old **Sanath Jayasuriya**, the 37-year-old left-handed batsman for **Sri Lanka**, walloping the hard ball rising off the slippery grass, sending seven shots over the fences, a home run to us, a six to cricket fans	/people/person/nationality	NA
Donskoi, only 36 years old, unknown outside of **Arkhangelsk** and perhaps better off for it, would stand little chance in a real campaign to be the leader of a country as sprawling, complex and deeply troubled as **Russia**	/location/location/contains	NA
In 1948, Rabbi Kret came to **New York City**, and with the help of Rabbi Yosef Eliyahu Henkin, of blessed memory, he was hired as the rabbi of the Old Broadway Synagogue in the West Harlem neighborhood of **Manhattanville**	/location/neighborhood /neighborhood_of	NA
But Justice Michael R. Ambrecht of State Supreme Court in Manhattan said that as a professional BASE -LRB- Bridge, Antenna, Span, Earth -RRB- jumper, Mr. Corliss, who has parachuted from the Eiffel Tower, the Golden Gate Bridge and the Petronas Towers in **Kuala Lumpur**, **Malaysia**, was experienced and careful enough to jump off a building without endangering his own life or anyone else's	/location/administrative_division /country	/people/person /nationality

extract such relation, whereas ours (**ete_pcnn_att**) leveraging the information of the corresponding entity types, which are severally 'person' and 'country', manage to extract it. Besides, the second and the third sentences may be analyzed similar to the first one. With respect to the fourth sentence, the model not employing the information of entity type is puzzled about the information of the context in the sentence, and has improperly identified the 'nationality' relation on the contrary. While our model correctly extract the relation.

5 Conclusion

In this paper, we propose a new relation extraction method that involves two techniques. We first present an entity type embedding module that integrates extra type information of subject and object entities to help eliminate word-sense ambiguity and emphasize the hidden relation between the two types of entities. We then design a new architecture of encoder based on GRU and residual network. The RPRN based encoder exploits the benefit of deeper neural network to learn the context of a sentence and extract finer syntactic and semantic features.

Experiment results show that both techniques improve the performance of relation extraction separately and they provide best result when used together. In the future work, we plan to apply the architecture to Semantic Role Labeling, Named Entity Recognition and so on, and then train the model of relation extraction and these tasks jointly to improve them together.

Acknowledgement. The authors from Huazhong University of Science and Technology, Wuhan, China, are supported by the Chinese university Social sciences Data Center (CSDC) construction projects (2017–2018) from the Ministry of Education, China. The first author, Dr. Yuming Wang, is a visiting scholar at the School of Computer Science, Georgia Institute of Technology, funded by China Scholarship Council (CSC) for the visiting period of one year from December 2017 to December 2018. Prof. Ling Liu's research is partially supported by the USA National Science Foundation CISE grant 1564097 and an IBM faculty award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

1. Zhang, Q., Chen, M., Liu, L.: A review on entity relation extraction. In: International Conference on Mechanical, Control and Computer Engineering, pp. 178–183 (2017)
2. Kumar, S.: A survey of deep learning methods for relation extraction. CoRR abs/1705.03645 (2017)
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 111–3119 (2013)
4. Leimeister, M., Wilson, B.J.: Skip-gram word embeddings in hyperbolic space. CoRR abs/1809.01498 (2018)

5. Shi, W., Gao, S.: Relation extraction via position-enhanced convolutional neural network. In: 2017 International Conference on Intelligent Environments (IE), pp. 142–148 (2017)
6. Ding, B., Wang, Q., Wang, B.: Leveraging text and knowledge bases for triple scoring: an ensemble approach - the Bokchoy triple scorer at WSDM Cup 2017. CoRR abs/1712.08356 (2017)
7. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL/IJCNLP (2009)
8. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6323, pp. 148–163. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15939-8_10
9. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L.S., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: ACL (2011)
10. Min, B., Grishman, R., Wan, L., Wang, C., Gondek, D.: Distant supervision for relation extraction with an incomplete knowledge base. In: HLT-NAACL (2013)
11. Goldwater, S.: Part of speech tagging. In: Encyclopedia of Machine Learning and Data Mining (2017)
12. Tan, Z., Wang, M., Xie, J., Chen, Y., Shi, X.: Deep semantic role labeling with self-attention. In: AAAI (2018)
13. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**, 436–444 (2015)
14. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING (2014)
15. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: EMNLP (2015)
16. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. CoRR abs/1601.00770 (2016)
17. Yao, K., Cohn, T., Vylomova, E., Duh, K., Dyer, C.: Depth-gated LSTM. CoRR abs/1508.03790 (2015)
18. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.T.: Cross-sentence N-ary relation extraction with graph LSTMs. TACL **5**, 101–115 (2017)
19. Song, L., Zhang, Y., Wang, Z., Gildea, D.: N-ary relation extraction using graph-state LSTM. In: EMNLP (2018)
20. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: ACL (2016)
21. Wang, L., Cao, Z., de Melo, G., Liu, Z.: Relation classification via multi-level attention CNNs. In: ACL (2016)
22. Du, J., Han, J., Way, A., Wan, D.: Multi-level structured self-attentions for distantly supervised relation extraction. In: EMNLP (2018)
23. Wu, Y., Bamman, D., Russell, S.J.: Adversarial training for relation extraction. In: EMNLP (2017)
24. Qin, P., Xu, W., Wang, W.Y.: DSGAN: generative adversarial training for distant supervision relation extraction. In: ACL (2018)
25. Dey, R., Salem, F.M.: Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1597–1600 (2017)
26. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: ICML (2013)

27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
28. Rohekar, R.Y.Y., Gurwicz, Y., Nisimov, S., Koren, G., Novik, G.: Bayesian structure learning by recursive bootstrap. CoRR abs/1809.04828 (2018)