REET: Joint Relation Extraction and Entity Typing via Multi-task Learning

Hongtao Liu[1], Peiyi Wang[1], Fangzhao Wu[2], Pengfei Jiao[3], Wenjun Wang[1]✉, Xing Xie[2], and Yueheng Sun[1]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{htliu, wangpeiyi979, wjwang, yhs}@tju.edu.cn
[2] Microsoft Research Asia, Beijing, China
wufangzhao@gmail.com, xingx@microsoft.com
[3] Center for Biosafety Research and Strategy, Tianjin University, Tianjin, China
pjiao@tju.edu.cn

Abstract. Relation Extraction (RE) and Entity Typing (ET) are two important tasks in natural language processing field. Existing methods for RE and ET usually handle them separately. However, relation extraction and entity typing have strong relatedness with each other, since entity types are informative for inferring relations between entities, and the relations can provide important information for predicting types of entities. Exploiting the relatedness between relation extraction and entity typing has the potential to improve the performance of both tasks. In this paper, we propose a neural network based approach to jointly train relation extraction and entity typing models using a multi-task learning framework. For relation extraction, we adopt a piece-wise Convolutional Neural Network model as sentence encoder. For entity typing, since there are multiple entities in one sentence, we design a couple-attention model based on Bidirectional Long Short-Term Memory network to obtain entity-specific representation of sentences. In our MTL frame, the two tasks share not only the low-level input embeddings but also the high-level task-specific semantic representations with each other. The experiment results on benchmark datasets demonstrate that our approach can effectively improve the performance of both relation extraction and entity typing.

Keywords: Relation Extraction · Entity Typing · Multi-task Learning.

1 Introduction

Relation Extraction (RE) is the task of extracting semantic relations between two entities from the text corpus. Entity Typing (ET) is a subtask of named entity recognition, which aims to assign types into the entity mention in a sentence. For example, given a sentence "Steve Jobs was the co-founder of Apple", entity typing aims to detect that the type of "Apple" is `Company` and the type of "Steve_Jobs" is `Person`, relation extraction aims to extract the `Co-Founder` relation between them. The two tasks both are important tasks in Natural Language Processing (NLP), which can be widely used in many applications such as Knowledge Base Completion, Question Answering and so on.

Various works have been proposed for relation extraction and entity typing. Most traditional works are feature-based methods. For example, Kambhatla et al. [5] combined diverse lexical, syntactic and semantic features of sentences and then employed maximum entropy model to extract relations. Recently some deep learning based methods about relation extraction and entity typing have been proposed. In relation extraction, for example Zeng et al. [19] adopted a convolutional neural network to represent sentences and used multi-instance learning to reduce the noise data during the distant supervision. Lin et al. [6] proposed a sentence-level attention model based on [19]. In entity typing task, Dong et al. [1] utilized recurrent neural network and multilayer perceptron to model sentences. Shimaoka et al. [14] introduced attention mechanism based on a BiLSTM model for entity typing classification.

These deep learning based methods have achieved better performance than those traditional works both in relation extraction and entity typing tasks. However, most existing works solve relation extraction and entity typing separately and regard them as independent tasks, which may be suboptimal. In fact, the two tasks have a strong inner relationship. For relation extraction, entity types are informative for inferring the semantic relations between entities. For example, the relation between "Steve_Jobs" (a person entity) and "Apple" (a company entity) would be related to position (e.g., co-founder) instead of uncorrelated ones (e.g., place_contains, friends). For entity typing task in turn, the relation information can guide the entity type classification. For example, the relation co-founder always exists in a person entity and a company entity. Hence, we can conclude that entity typing and relation extraction can provide helpful information for each and this correlation between them should be fully exploited, which could benefit for two tasks. Nevertheless, most works regard them as separate tasks and ignore the rich connection information.

Motivated by above observations, in this paper we propose a neural multi-task learning framework REET for joint Relation Extraction and Entity Typing. Specifically, we develop a relation extraction model based on PCNN [19]. For entity typing task, considering that there are multiple entities in one sentence, we design a novel couple-attention architecture based on Bidirectional Long Short-Term Memory (BiLSTM), which can extract the semantic information of different entities in one sentence. To characterize the connections between relation extraction and entity typing, in our framework the two tasks share two-level information: (1) the low-level word embeddings in the input layer, (2) the high-level task-specific semantic representations obtained from all tasks. In our framework, both tasks can gain better generalization capabilities via integrating the domain-specific information from related tasks. We evaluate our approach on two benchmark datasets and the experiment results show that our approach can effectively improve the performance of both relation extraction and entity typing.

2 Related Work

2.1 Relation Extraction

Many relation extraction methods have been proposed these years. Traditional works mainly utilized human-designed lexical and syntactic features e.g., POS tagging, shortest dependency path to extract relations [2, 5]. Recently, some deep learning based methods have been proposed and outperformed those traditional feature-based methods a lot. Zeng et al. [19] encoded sentences via convolutional neural networks and utilized multi-instance learning as sentence selector to reduce noise data in distant supervision. Lin et al. [6] introduced sentence-level attention among sentences to alleviate the noise sentences based on [19]. Some more complicated methods have been proposed recently. Ye et al. [17] explored the class ties (e.g., inner interaction among relations) and proposed a general pairwise ranking framework to learn this association between relations. Liu et al. [7] adopted Sub-Tree Parse to remove noisy words that are irrelevant to relations and initialized their model with parameters learned from the entity classification by transfer learning. In this paper, We utilize the basic model PCNN [19] as our sentence encoder and selector for the relation extraction task.

2.2 Entity Typing

Entity Typing is a subtask of named entity recognition. Traditional methods rely heavily on handcrafted features [9]. With the development of deep learning, more and more neural network methods have been proposed [1, 14, 13] and achieve significant improvement. Dong et al. [1] adopted a neural architecture that combined fully-connected layers and recurrent layers to model sentence and entity. Shimaoka et al. [13] further applied attention mechanism in recurrent neural networks. These models are all designed for the problem that there is only one entity mention in a sentence, however there are two entities in relation extraction scenario. Hence we design a novel couple-attention neural network model based on Bidirectional Long Short-Term Memory (BiLSTM), which takes the information of both entities into attention mechanism.

2.3 Multi-Task Learning

Multi-Task Learning (MTL) can improve the performance of related tasks by leveraging useful information among them and can reduce the risk of overfitting and generalize better on all tasks [11]. Hence, we propose a neural multi-task learning framework for relation extraction and entity typing, and incorporate them via sharing two-level parameters, which can characterize the task-specific information and connection information between tasks simultaneously.

3 Methodology

In this section, we describe our multi-task learning framework in details. We will give the definitions of relation extraction and entity typing first and then introduce models respectively. Afterwards, we integrate the two models jointly via a multi-task learning framework. The overview architecture of our approach is shown in Fig. 1.

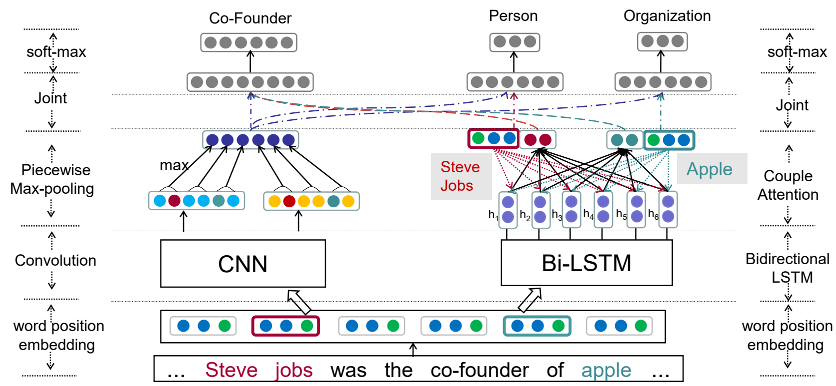

Fig. 1. Overview architecture of our model via Multi-task Learning. There are two joint shared parameters between Relation Extraction and Entity Typing: (a) the low-level input embedding. (b) the high-level feature vectors integration respectively.

3.1 Problem Definition

Given a sentence $s = \{w_1, w_2, \cdots, e_1, \cdots, e_2 \cdots\}$ and two target entities (e_1, e_2), relation extraction task is to predict which relation could exist between e_1 and e_2, and entity typing task aims to assign categories to e_1 and e_2 respectively. As a result, there are three subtasks in our multi-task learning scenario: relation extraction for the entity pair, entity typing for e_1 and entity typing for e_2.

3.2 Relation Extraction Module

In this paper, we develop a basic relation extraction model based on PCNN [19] in the relation extraction task.

Word Embeddings For a sentence $s = \{w_1, w_2, \cdots, w_n\}$, we transform each word w_i into a low-dimensional real-valued vector $\mathbf{w}_i \in \mathbf{R}^{d_w}$, where d_w is the dimension of word embedding vectors.

Position Embeddings Position Feature (PF) has been widely used in RE, which encodes the relative distances between each word and the two entities into low-dimensional vectors as position embedding for each word. We concatenate the word embedding and position embedding as the final representation of each word: $\mathbf{w}_i \in \mathbf{R}^{d_w+2*d_p}$, where d_p is the dimension of position embedding.

Convolution and Piece-wise Max-Pooling Each sentence can be represented as a matrix: $\mathbf{s} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]$, and we will conduct the convolution operations to extract the semantic features of sentences. Given K convolution kernels denoted as $\mathbf{F} = \{\mathbf{f_1}, \mathbf{f_2}, \cdots, \mathbf{f_K}\}$, the window size of kernels l, the convolutional operation between the i-th kernel and the j-th window of input words is defined as:

$$c_{i,j} = \mathbf{f_i} \odot \mathbf{w}_{j:j+l-1} \ ,$$

where $\odot$ is the inner-product operation. After stacking all windows, the output of the convolutional layer will be a set of vectors $\mathbf{C} = [\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_K}]$, $\mathbf{c_i} = [c_{i,1}, c_{i,2}, \cdots, c_{i,n}]$ and n is the sequence length.

In pooling phase, we adopt piece-wise max-pooling following [20], each sentence is divided into three segments by two entities, then we conduct max-pooling in each segment of the i-th kernel:

$$p_{i,j} = \max\left(\mathbf{c_{i,j}}\right) \ \ 1 \leq i \leq K, j = 1, 2, 3 \ .$$

As a result, we will obtain a 3-dimensional vector for each kernel, after stacking all kernels, we will get the pooling result: $\mathbf{z} = [p_{i1}, p_{i2}, p_{i3}]_{i=1}^{K} \in \mathbf{R}^{3\times K}$. After that, we apply a non-linear function e.g., hyperbolic tangent to denote the final fixed-length sentence representation $\mathbf{S} \in \mathbf{R}^{3\times K}$:

$$\mathbf{S} = \tanh(\mathbf{z}) \ .$$

3.3 Entity Typing Module

In relation extraction scenario, there are multiple entities in one sentence, while the previous entity typing tasks focus on the sentence with only one entity. To address this issue, we design a couple-attention Bidirectional Long Short-Term Memory model. Two entity typing tasks share the BiLSTM layer and utilize the entity-specific attention vectors to distinguish the different entities in the attention layer as illustrated in the right part of Fig. 1. Note that our model can handle situation when there are multi entities (larger than 2) in a sentence.

Shared BiLSTM Sentence Encoder Long Short-Term Memory (LSTM) is capable of learning long-term dependencies in sentences So here we use the Bidirectional LSTM (BiLSTM) [12] networks, i.e., there are two sub LSTM networks for the sentences, and one is for forward pass from left to right and another is for backward pass in an opposite direction from right to left. Given the input sequence $\mathbf{s} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]$, the formula for the BiLSTM unit is denoted as:

$$\begin{aligned}
\overrightarrow{\mathbf{h_i}} &= \overrightarrow{\mathtt{LSTM}}(\mathbf{w_t}) \ \ , \ t \in [1, n] \ , \\
\overleftarrow{\mathbf{h_i}} &= \overleftarrow{\mathtt{LSTM}}(\mathbf{w_t}) \ \ , \ t \in [n, 1] \ , \\
\mathbf{h_i} &= [\overrightarrow{\mathbf{h_i}}, \overleftarrow{\mathbf{h_i}}] \ .
\end{aligned}$$

Then $\mathbf{h_i}$ denotes the high-level semantic representation of the i-th word, which will be shared for the two entity typing tasks.

Couple-Attention A word could be of different information in terms of different entities and should not be treated equally. Hence, we introduce a couple-attention mechanism to get entity-related representations for sentences. Specifically, we regard the word embeddings of entity mentions as query vectors in attention layer. Hence, for each entity e_m, we can denote the entity-related sentence representation as:

$$\begin{aligned} k_i &= \tanh(W_s\mathbf{h_i} + b_s) \ , \\ \alpha_m^i &= \frac{\exp\left(k_i^T \mathbf{e_m}\right)}{\sum_j^n \exp\left(k_j^T \mathbf{e_m}\right)} \ , \ m = 1, 2 \ , \\ \mathbf{v_m} &= \sum_i^n \alpha_m^i \mathbf{h_i} \ , \ m = 1, 2 \ , \end{aligned}$$

where W_s is the parameter matrix of attention, $\mathbf{e_m}$ i.e., $\mathbf{e_1}, \mathbf{e_2}$ are the word embeddings of two entities in sentence s, and α_m^i indicates the weights of the i-th word under the m-th entity, $\mathbf{v_m}$ are the sentence feature vector for the m-th entity. In order to get more entity-specific information, we concatenate the entity embedding with the output above, hence, the final feature vectors for two entity typing tasks are:

$$\mathbf{T_1} = [\mathbf{e_1}, \mathbf{v_1}] \ , \quad \mathbf{T_2} = [\mathbf{e_2}, \mathbf{v_2}] \ .$$

3.4 Multi-Task Learning Framework

In this part we will introduce the multi-task learning framework aiming at how to combine the relation extraction and the entity typing together. In multi-task learning, which module to share is crucial; according to [15], in most NLP tasks, sharing representations at lower-level layers is necessary and effective. Hence, we first share the input layer i.e., the word representations of a sentence: $\mathbf{s} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]$ between relation extraction task and entity typing task. Besides, considering that the high-level semantic representations of other tasks can be a feature augmentation for the current task, we further integrate the feature vectors for relation extraction tasks and entity typing tasks.

As shown in Fig. 2, we implement two typical MTL models REET-1 and REET-2 according to which modules to share between two tasks.

REET-1 The relation extraction task and entity typing task independently only share input embedding layers. After obtaining feature vectors for all tasks, i.e., $\mathbf{S}, \mathbf{T_1}, \mathbf{T_2}$, we adopt soft-max layer to calculate the confident probability of all labels in each task:

$$\begin{aligned} \mathbf{p_r} &= \mathtt{softmax}(W_r\mathbf{S} + b_r) \ , \\ \mathbf{p_{t_i}} &= \mathtt{softmax}(W_{t_i}\mathbf{T_i} + b_{t_i}) \ , \ i = 1, 2 \ , \end{aligned}$$

where $\mathbf{p_r}$ and $\mathbf{p_{t_i}}$ are the prediction probabilities for RE and ET respectively.

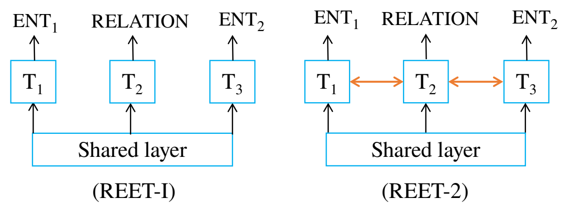

Fig. 2. Two MTL Architectures. T_1, T_2, T_3 are task-specific modules: Entity Typing for e_1, Relation Extraction,Entity Typing for e_2.

REET-2 In order to further explore the deep interaction between RE and ET, we design REET-2 to share more task-specific information. Specifically, we concatenate the feature vectors of relation and entity types before the last classification layer, denoted as:

$$\mathbf{p_r} = \mathtt{softmax}(W_r[\mathbf{T_1}, \mathbf{S}, \mathbf{T_2}] + b_r) \ ,$$
$$\mathbf{p_{t_i}} = \mathtt{softmax}(W_{t_i}[\mathbf{S}, \mathbf{T_i}] + b_{t_i}) \ , \ i = 1, 2 \ .$$

Hence, in this way, the relation extraction task and entity typing tasks can share a high-level feature with each, which indicate the task-specific information.

Training Objective and Optimization For each task, we define the loss function via cross entropy:

$$L_r(\theta_0) = -\frac{1}{R}\sum_{k=1}^{R} y_r \log \mathbf{p_r}(k) \ , \qquad L_{\mathbf{t_i}}(\theta_i) = -\frac{1}{C}\sum_{k=1}^{C} y_{\mathbf{t_i}} \log \mathbf{p_{t_i}}(k) \ , \ i = 1, 2 \ ,$$

where R, C are the number of relation and entity types respectively. y_r and $y_{\mathbf{t_i}}$ are the true class labels for relation extraction and entity type tasks and $\theta = \{\theta_0, \theta_1, \theta_2\}$ covers all the parameters in our model.

We combine all three function in a weighted sum way as our final loss function:

$$L(\theta) = \lambda L_{t_1} + \lambda L_{t_2} + (1 - \lambda) L_r \ ,$$

where $0 \leq \lambda \leq 1$ denotes the balance weight for the loss of entity typing tasks. In the training phase, we adopt Adadelta [18] to optimize the objective $L(\theta)$.

4 Experiments

4.1 Experimental Settings

Dataset Our experiments are performed on NYT+Freebase and Google Distant Supervision (GDS) datasets.

NYT+Freebase: The dataset is built by [10] and generated by aligning entities and relations in Freebase with the corpus of New York Times (NYT). The articles of NYT from year 2005-2006 are used as training data, and articles from 2007 are used as testing data. We extract the field `type.object.type` from

Freebase as entities types, including five domain types: `person,location,art ,organization` and `other`.

GDS: This dataset is newly built by [4] which is extracted from Google Relation Extraction corpus[4]. Different from NYT+Freebase, GDS is a human-judged dataset and each entity-pair in the dataset is judged by at least 5 raters. So the labelled relation is correct for each instance set in GDS.

The statistics of the two datasets is summarized in Table 1.

Dataset	# relations	#entity types	# sentences	# entity-pair
NYT+Freebase Dataset				
Train	53	5	455,771	233,064
Dev	53	5	114,317	58,635
Test	53	5	172,448	96,678
GDS Dataset				
Train	5	25	11,297	6,498
Dev	5	25	1,864	1,082
Test	5	25	5,663	3,247

Table 1. Statistics of NYT+Freebase and GDS datasets.

Evaluation Metrics In relation extraction task, following previous works [19, 6, 7], we evaluate the results with held-out metrics, which can provide the approximate precision about the relations extracted by the models. We will report the Precision-Recall Curve and the Precision@N (P@N) in the held out evaluation. In entity typing task, we will show the classification metrics F1-score to evaluate our approach.

Hyper Parameter Settings We explore different combination of hyper parameters using the validation datasets in experiments. The best parameter configuration is loss balance weight $\lambda = 0.6$, BiLSTM hidden size $h = 50$, the embedding dimensions $d_w = 50$ and $d_p = 5$, the filter number and window size in CNN $K = 230$ and $l = 3$ respectively.

4.2 Performance in Relation Extraction

In this section, we will investigate the performance of our MTL framework in relation extraction task.

Baseline Methods We list some recent competitive methods as baselines.

Traditional feature-based methods:

Mintz [8] designed various features for all sentences to extract semantic relations.

MultiR [3] adopted multi-instance learning in distant supervision relation extraction.

MIMLRE [16] regarded RE as a multi-instance and multi-label problem in a feature-based method.

[4] https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html

Recently neural network based methods:

PCNN [19] utilized the convolutional neural network as sentence encoder and used multi-instance learning to select one sentence for one entity pair.

PCNN+ATT [6] proposed a sentence-level attention based on PCNN to alleviate the wrong labeling problem.

BGWA [4] proposed a word and sentence attention model based on BGRU to capture the important information in distant supervision.

BGRU+STP+EWA+TL [7] (abbreviation as **BGRU+ALL**), which is a joint model as well, utilized Sub-Tree Parse (STP), Entity Word-level Attention (EWA) and incorporated entity type information via pre-train transfer learning (TL).

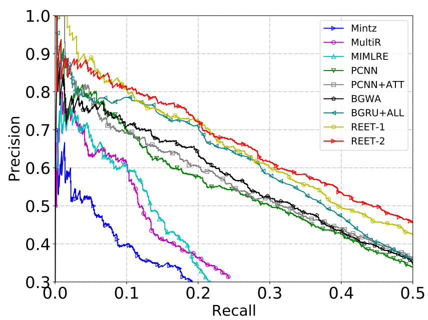

Fig. 3. Precision-Recall curves on NYT.

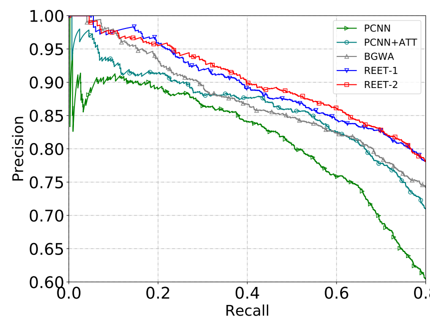

Fig. 4. Precision-Recall curves on GDS.

Performance Evaluation The Precision-Recall curves on NYT+Freebase and GDS[5] are presented in Fig 3 and Fig 4 respectively. Most methods performs much better on GDS than on NYT+Freebase. The reason is that there are only five categories in GDS and the labelled relation in GDS is correct and without noise. From the results, we can observe that (1) REET-1 and REET-2 both outperform PCNN quite a lot, which shows the advantages of our multi-task learning method and indicates the entity typing task can indeed boost the performance of relation extraction task. (2) REET-1 and REET-2 can both outperform than BGRU+ALL on NYT+Freebase, which utilized entity type information as side information via transfer learning. This is because our multi-task learning method can exploit not only the low-level but also the high-level interaction between RE and ET, while BGRU+ALL using the entity type for RE can not make full use of the complementary information of RE and ET. (3) REET-1 and REET-2 achieve the best performance along the entire curves on the two datasets, which indicates that the entity typing task in our MTL frame can be beneficial for relation extraction task. Besides, REET-2 performs slightly better than REET-1. The reason is that in REET-2, relation extraction

[5] On GDS dataset we only compare with some recent baselines since the dataset is newly released in year 2018

P@N (%)	Mintz	MultiR	MIML	PCNN	PCNN+ATT	BGWA	BGRU+ALL	REET-1	REET-2
100	52.7	69.4	71.1	78.7	81.8	82.0	87.0	**88.3**	87.8
200	50.2	65.1	63.8	72.8	71.1	75.0	83.0	83.2	**83.5**
300	46.9	62.0	61.1	67.8	69.3	72.0	78.0	78.0	**79.2**
AVG	49.4	65.6	65.3	73.1	74.1	78.4	82.7	83.2	**83.6**

Table 2. Precision@Top K on NYT+Freebase dataset.

task integrates a high-level representation of entity typing, which is an explicit feature augmentation for the relation classifier.

In addition, following previous works, we evaluate our models using P@N metric in held-out evaluation on NYT+Freebase dataset shown in Table 2. We can find that the conclusion is consistent with the PR-Curves above, and our REET-1 and REET-2 achieve the best P@N values.

4.3 Performance in Entity Typing

Next, we will investigate the performance of our MTL models in entity typing task. Here we will compare two baseline methods in entity typing:

BiLSTM [14]: a BiLSTM model for entity typing classification with attention mechanism.

BiLSTM+Co_Att: our proposed Couple-Attention BiLSTM model in single mode i.e., without relation extraction task.
As there are multiple entities in one sentence, in experiments we will report the average metrics of the entities.

F1 (%)	BiLSTM	BiLSTM+Co_ATT	REET-1	REET-2
NYT+Freebase	94.7	95.5	96.5	**96.8**
GDS	70.1	72.8	74.2	**76.6**

Table 3. Classification performance of entity typing task.

The result is shown in Table 3 and the difference of baselines to REET-1 and REET-2 is statistically significant at 0.05 level. We can conduct that (1) the multi-task learning methods REET-1 and REET-2 outperform than BiLSTM and Co-Att both under single task mode. This indicates that relation extraction task can provide semantically information for entity typing task in our MTL framework. (2) BiLSTM+Co_Att performs better than BiLSTM [14], which shows that the effectiveness of our couple-attention mechanism in entity typing. The reason is that Couple-Attention can utilize more entity-specific information for each entity in a sentence. (3) REET-2 achieves a better results compared to REET-1. It is consistent with the conclusion in the relation extraction experiment and illustrates that a high-level integration will be beneficial for all the tasks.

4.4 Parameter Analysis

In this section, we explore the influence of balance weight parameter λ, which controls the importance of entity typing task. We report the average micro F1-

score of two entities in entity typing task and the average value of P@N in relation extraction task with NYT+Freebase dataset.

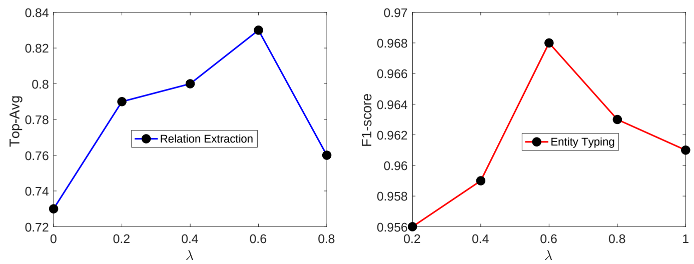

Fig. 5. the influence of parameter of λ in relation extraction and entity typing.

As shown in Fig 5, we can conclude that the influence of parameter λ shows similar patterns in the two tasks. As λ increases, the performance of the two tasks first increases, then reaches the peak, and decreases afterwards. The reason is that when λ is too small, for relation extraction the entity type information could not be used fully, and for entity typing the model would update very slowly. Hence the performances of two tasks are poor. However, when the λ becomes too large, the information of entity typing will be overemphasized and the relation information will be ignored, which leads to a poor performance as well. We can find that when the value λ is about 0.6, both relation extraction and entity typing can achieve the best performance.

5 Conclusion

In this paper, we propose a multi-task learning frame that integrates relation extraction task and entity typing task jointly since relation extraction and entity typing have strong relatedness with each other to be utilized. We develop a relation extraction model based on PCNN, and design a couple-attention BiLSTM model for entity typing task fit for multiple entities in a sentence. The two tasks share low-level (i.e., input embedding layer) and high-level information (i.e., task-specific feature), and in this way, the rich relatedness of RE and ET can be exploited fully. Extensive experimental results on two benchmark datasets validate the effectiveness of our multi-task learning frame, and both relation extraction task and entity typing task achieve a significant improvement and our approach outperforms many baseline methods.

Acknowledgement

This work was supported by the National Key R&D Program of China (2018YFC0831005), the Science and Technoloy Key R&D Program of Tianjin (18YFZCSF01370) and the National Social Science Fund of China (15BTQ056).

References

1. Dong, L., Wei, F., Sun, H., Zhou, M., Xu, K.: A hybrid neural model for type classification of entity mentions. In: IJCAI. pp. 1243–1249 (2015)
2. GuoDong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: ACL. pp. 427–434. Association for Computational Linguistics (2005)
3. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: ACL. pp. 541–550 (2011)
4. Jat, S., Khandelwal, S., Talukdar, P.: Improving distantly supervised relation extraction using word and entity based attention. vol. abs/1804.06987 (2018)
5. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: ACL. p. 22. Association for Computational Linguistics (2004)
6. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: ACL. vol. 1, pp. 2124–2133 (2016)
7. Liu, T., Zhang, X., Zhou, W., Jia, W.: Neural relation extraction via inner-sentence noise reduction and transfer learning. arXiv preprint arXiv:1808.06738 (2018)
8. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings ofthe Joint Con- ference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language. pp. 1003–1011 (2009)
9. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: CoNLL. pp. 147–155 (2009)
10. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: ECML. pp. 148–163. Springer (2010)
11. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
12. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
13. Shimaoka, S., Stenetorp, P., Inui, K., Riedel, S.: An attentive neural architecture for fine-grained entity type classification. NAACL pp. 69–74 (2016)
14. Shimaoka, S., Stenetorp, P., Inui, K., Riedel, S.: Neural architectures for fine-grained entity type classification. EACL **1**, 1271–1280 (2017)
15. Søgaard, A., Goldberg, Y.: Deep multi-task learning with low level tasks supervised at lower layers. In: ACL. vol. 2, pp. 231–235 (2016)
16. 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 (2012)
17. Ye, H., Chao, W., Luo, Z., Li, Z.: Jointly extracting relations with class ties via effective deep ranking. meeting of the Association for Computational Linguistics pp. 1810–1820 (2017)
18. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
19. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: EMNLP. pp. 1753–1762 (2015)
20. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING. pp. 2335–2344 (2014)