Exploiting Entity BIO Tag Embeddings and Multi-task Learning for Relation Extraction with Imbalanced Data

Wei Ye[1]*†, **Bo Li**[1,3]*, **Rui Xie**[1,2], **Zhonghao Sheng**[1,2],
Long Chen[1,2] and **Shikun Zhang**[1]
[1]National Engineering Research Center for Software Engineering, Peking University
[2]School of Software and Microelectronics, Peking University
[3]Automation Dept, Beijing University of Posts and Telecommunications.
wye@pku.edu.cn, deepblue.lb@gmail.com,
{ruixie, zhonghao.sheng, clcmlxl, zhangsk}@pku.edu.cn

Abstract

In practical scenario, relation extraction needs to first identify entity pairs that have relation and then assign a correct relation class. However, the number of non-relation entity pairs in context (negative instances) usually far exceeds the others (positive instances), which negatively affects a model's performance. To mitigate this problem, we propose a multi-task architecture which jointly trains a model to perform relation identification with cross-entropy loss and relation classification with ranking loss. Meanwhile, we observe that a sentence may have multiple entities and relation mentions, and the patterns in which the entities appear in a sentence may contain useful semantic information that can be utilized to distinguish between positive and negative instances. Thus we further incorporate the embeddings of character-wise/word-wise BIO tag from the named entity recognition task into character/word embeddings to enrich the input representation. Experiment results show that our proposed approach can significantly improve the performance of a baseline model with more than 10% absolute increase in F1-score, and outperform the state-of-the-art models on ACE 2005 Chinese and English corpus. Moreover, BIO tag embeddings are particularly effective and can be used to improve other models as well.

1 Introduction

Relation extraction, which aims to extract semantic relations from a given instance—entity pair and the corresponding text in context, is an important and challenging task in information extraction. It serves as a step stone for many downstream tasks such as question answering and knowledge graph construction.

Traditionally, researchers mainly use either feature-based methods (Kambhatla, 2004; Boschee et al., 2005; GuoDong et al., 2005; Jiang and Zhai, 2007; Chan and Roth, 2010; Sun et al., 2011; Nguyen and Grishman, 2014) or kernel-based methods (Zelenko et al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Mooney and Bunescu, 2006; Zhang et al., 2006; Zhou et al., 2007; Giuliano et al., 2007; Qian et al., 2008; Nguyen et al., 2009; Sun and Han, 2014) for relation extraction, which tend to heavily rely on handcraft features and existing natural language processing (NLP) tools. Recently, deep learning models, including convolutional neural network (CNN) (Liu et al., 2013; Zeng et al., 2014; Nguyen and Grishman, 2015; Zeng et al., 2015; dos Santos et al., 2015; Lin et al., 2016) and recurrent neural network (RNN) (Miwa and Bansal, 2016; Zhou et al., 2016; She et al., 2018) w/o variants of attention mechanism have been widely applied to relation extraction and achieved remarkable success.

The relation extraction task can be divided into two steps: determining which pair of entities in a given sentence has relation, and assigning a correct relation class to the identified entity pair. We define these two steps as two related tasks: **Relation Identification** and **Relation Classification**. If one only needs to categorize the given entities that are guaranteed to have some expected relation, then relation extraction is reduced to relation classification (Nguyen and Grishman, 2015).

One variation of relation classification is the introduction of a new artificial relation class "Other." If the number of non-relation entity pairs in context (**negative instances**) in the dataset is comparable to the number of entity pairs that have relation in context (**positive instances**), then the non-relation pairs can be treated as having the relation class Other.

* indicates equal contribution.
† Corresponding author.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1351–1360
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

Strictly speaking, most existing studies of relation extraction treat the task as relation classification. However, relation extraction often comes with an extremely imbalanced dataset where the number of non-relation entity pairs far exceeds the others, making it a more challenging yet more practical task than relation classification. For example, after filtering out those entity pairs whose entity type combination has never appeared in the Chinese corpus of ACE 2005, there are still more than 200,000 entity pairs left, in which the positive/negative instance ratio is about 1:20. In this paper, we focus on the relation extraction task with an imbalanced corpus, and adopt multi-task learning paradigm to mitigate the data imbalance problem.

Only a few studies have considered the negative effect of having too many negative instances. Nguyen and Grishman (2015) proposed using CNN with filters of multiple window sizes. dos Santos et al. (2015) focused on learning the common features of the positive instances by computing only the scores of the relation classes excluding the class Other, and proposed using a pairwise ranking loss. We have also adopted these methods in our approach.

For relation classification, the prediction error can be categorized into three types: 1) false negative—predicting a positive instance to be negative; 2) false positive—predicting a negative instance to be positive; 3) wrong relation class—predicting a positive instance to be positive yet assigning a wrong relation class. After training a baseline model to perform relation classification on the extremely imbalanced ACE 2005 Chinese corpus and dissecting its prediction errors, we find that the proportion of these three types of error are 30.20%, 62.80% and 7.00% respectively. It is conceivable that to improve a model's performance on such corpus, it is best to focus on telling positive and negative instances apart.

Since the negative instances may not have much in common, distinguishing between positive and negative instances is much more challenging than only classifying positive instances into a correct class. Moreover, the total number of positive instances combined is more comparable to the number of negative instances than positive instances of any individual relation class alone. Based on these rationales, we propose to jointly train a model to do another binary classification task—relation identification—alongside relation classification to mitigate the data imbalance problem.

Another facet that most existing studies fail to consider is that there may be multiple relation mentions in a given sentence if it contains multiple entities. In the Chinese corpus of ACE 2005, there are 4.9 entities and 1.34 relation mentions in a sentence on average. The patterns in which these entities appear in the sentence can provide useful semantic information to distinguish between positive and negative instances. Therefore, we exploit the character-wise/word-wise BIO (Beginning, Inside, Outside) tag used in the named entity recognition (NER) task to enrich the input representation. The details of our approach will be presented in Section 2.

We conducted extensive experiments on ACE 2005 Chinese and English corpus. Results show that both the novel multi-task architecture and the incorporation of BIO tag embeddings can improve the performance, and the model equipped with both achieves the highest F1-score, significantly outperforming the state-of-the-art models. Analysis of the results indicates that our proposed approach can successfully address the problem of having a large number of negative instances.

To summarize, we make the following contributions in this paper:

1. We propose a multi-task architecture which jointly trains a model to perform relation identification with cross-entropy loss and relation classification task with ranking loss, which can successfully mitigate the negative effect of having too many negative instances.

2. We incorporate the embeddings of character-wise/word-wise BIO tag from NER task to enrich the input representation, which proves to be very effective not only for our model but for other models as well. We argue that BIO tag embeddings could be a general part of character/word representation, just like the entity position embeddings (Zeng et al., 2014) that many researchers would use in recent years.

2 Proposed Approach

We have designed a novel multi-task architecture which combines two related tasks: 1) relation identification, which is a binary classification problem to determine whether a given entity pair has relation; 2) relation classification, which is a multiple classification problem to determine the relation class. Figure 1 shows the overall architecture.

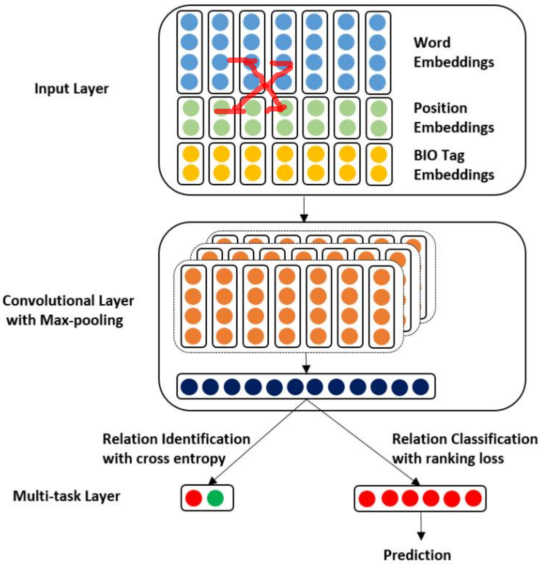

Input Layer
Word Embeddings
Position Embeddings
BIO Tag Embeddings
Convolutional Layer with Max-pooling
Relation Identification with cross entropy
Relation Classification with ranking loss
Multi-task Layer
Prediction

Figure 1: The overall multi-task architecture. To demonstrate, there are three window sizes for filters in the convolutional layer, as denoted by the three-layer stack; for each window size there are four filters, as denoted by the number of rows in each layer. Max-pooling is applied to each row in each layer of the stack, and the dimension of the output is equal to the total number of filters.

Three are three main parts in the architecture:

- **Input Layer** Given an input sentence x of n words[1] $\{x_1, x_2, ..., x_n\}$ with m entities $\{e_1, e_2, ..., e_m\}$ where $e_i \in \boldsymbol{x}$, and two target entities $e_{t1}, e_{t2} \in \{e_1, e_2, ..., e_m\}$, the input layer transforms the sentence into a matrix $\boldsymbol{X}$, which includes word embeddings, position embeddings and BIO tag embeddings of each word.

- **Convolutional Layer with Max-pooling** Following the input layer is a convolutional layer that extracts high-level features, with filters (convolution kernels) of multiple window sizes (Nguyen and Grishman, 2015). Then max-pooling is applied to each feature map to reduce dimensionality.

- **Multi-task Layer** In the multi-task layer, the model jointly learns the relation identification task using cross-entropy loss and the relation classification task using ranking loss.

2.1 Input Layer

- **Word Embeddings** We use word embeddings with random initialization for each word in the input sentence. The dimension of word embeddings is d^w.

- **Position Embeddings** We also employ position embeddings to encode the relative distance between each word and the two target entities in the sentence. We believe that more useful information regarding the relation is hidden in the words closer to the target entities. The dimension of position embeddings is d^p.

- **BIO Tag Embeddings** Since an input sentence often contains more than two entities, we utilize the BIO tag information of entities to enrich the representation of the input. More specifically, for each word in the input sentence, if the word is part of an entity, we use the entity type T to label the start of the entity as B_T, and label the rest of the entity as B_I. If the word is not part of an entity, then we label the word as O. The dimension of BIO tag embeddings is d^t.

After concatenating all three embeddings together for each word, we transform a sentence into a matrix $\boldsymbol{X} = [\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_n}]$ as the input representation, where the column vector $\boldsymbol{w_i} \in \mathbb{R}^{d^w+2*d^p+d^t}$. Figure 2 illustrates how to derive position embeddings and BIO tag embeddings.

2.2 Convolutional Layer with Multi-Sized Window Kernels

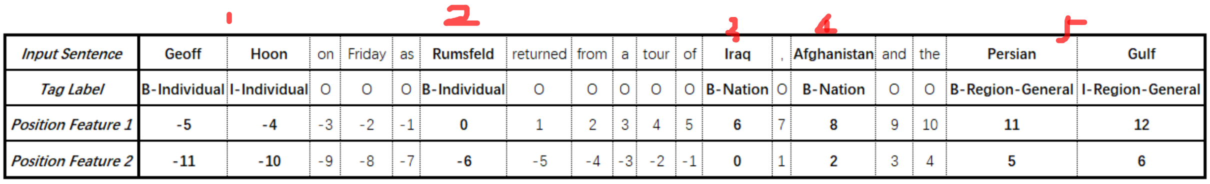
Next, the matrix $\boldsymbol{X}$ is fed into the convolutional layer to extract high-level features. A filter with window size k can be denoted as $\boldsymbol{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, .., \boldsymbol{f_k}]$, where the column vector $\boldsymbol{f_i} \in \mathbb{R}^{d^w+2*d^p+d^t}$. Apply the convolution operation on the two matrices $\boldsymbol{X}$ and $\boldsymbol{F}$, and we get a score sequence $T = \{t_1, t_2, ..., t_{n-k+1}\}$:

$$t_i = g(\sum_{j=0}^{k-1} \boldsymbol{f}_{\boldsymbol{j+1}}^{T} \boldsymbol{w}_{\boldsymbol{j+i}} + b) \qquad (1)$$

where g is a non-linear function and b is bias.

In our experiments, we apply zero-paddings during the convolution operation, so that the score

[1] We use character-wise model for Chinese corpus and word-wise model for English corpus. For simplicity sake, we use "word" to denote either an English word or a Chinese character to present our model.



Input Sentence	Geoff	Hoon	on	Friday	as	Rumsfeld	returned	from	a	tour	of	Iraq	,	Afghanistan	and	the	Persian	Gulf
Tag Label	B-Individual	I-Individual	O	O	O	B-Individual	O	O	O	O	O	B-Nation	O	B-Nation	O	O	B-Region-General	I-Region-General
Position Feature 1	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12
Position Feature 2	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6

Figure 2: Illustration of BIO tag information and positional information for a given instance. In this example, there are five entities in the input sentence, and the target entities are the second and the third.

sequence has the same length as the input sequence, which is n, instead of $n - k + 1$ if we apply Equation 1 which assumes no padding.

There are multiple filters with different window sizes in the convolutional layer. Then max-pooling is applied to the outputted feature map of each filter. Eventually the input sentence x is represented as a column vector $\boldsymbol{r}$ with a dimension that is equal to the total number of filters.

2.3 Multi-Task Layer

- **Relation Identification with Cross-entropy Loss** For the binary classification task of relation identification we use cross-entropy loss. Positive instances are labelled "1" and negative instances "0."

  If p is the one-hot true distribution over all classes $C = \{c\}$ and q is the distribution a model predicts, then the cross-entropy loss of a given instance can be defined as follows:

$$H(p, q) = -\sum_{c \in C} p(c) log(q(c)) \tag{2}$$

  So the loss of this task can be defined as:

$$loss_1 = -\sum (p(1) log(q(1)) + p(0) log(q(0))) \tag{3}$$

- **Relation Classification with Ranking Loss** For the multiple classification task of relation classification, we use the pairwise ranking loss proposed by (dos Santos et al., 2015).

  Given the sentence representation $\boldsymbol{r}$, the score for class c is computed as:

$$s_c = \boldsymbol{r}^T [\boldsymbol{W}^{classes}]_c \tag{4}$$

  where $\boldsymbol{W}^{classes}$ is a matrix to be learned, whose number of columns is equal to the number of classes. $\boldsymbol{W}_c^{classes}$ is a column vector corresponding to class c, whose dimension is equal to that of $\boldsymbol{r}$.

  For each instance, the input sentence x has a correct class label y^+ and incorrect ones y^-. Let s_{y^+} and s_{y^-} be the scores for y^+ and y^- respectively, then the ranking loss can be computed by the following two equations:

$$L^+ = log(1 + exp(\gamma(m^+ - s_{y^+}))) \tag{5}$$

$$L^- = log(1 + exp(\gamma(m^- + s_{y^-}))) \tag{6}$$

  where m^+ and m^- are margins and γ is a scaling factor. L^+ decreases as the score s_{y^+} increases, and is close to zero when $s_{y^+} > m^+$, which encourages the network to give a score greater than m^+ for the correct class. Similarly, L^- decreases as the score s_{y^-} decreases, and is close to zero when $s_{y^-} < -m^-$, which encourages the network to give scores smaller than $-m^-$ for incorrect classes.

  For the class Other, only L^- is calculated to penalize the incorrect prediction. And following (dos Santos et al., 2015), we only choose the class with the highest score among all incorrect classes as the one to perform a training step. Then we optimize the pairwise ranking loss function:

$$loss_2 = \sum (L^+ + L^-) \tag{7}$$

The total loss function for multi-task training is:

$$L = \alpha \cdot loss_1 + \beta \cdot loss_2 \tag{8}$$

where α and β are weights of the two losses. In our experiments, we find that $\alpha = \beta$ yields the best result.

2.4 Prediction

We only use the class score s_c in the multiple classification task to make predictions, while the binary classification task is only used for optimizing the network parameters.

Given an instance, the prediction P is made by:

$$P = \begin{cases} \arg\max_{c}(s_c) & max(s_c) \geq \theta \\ Other & max(s_c) < \theta \end{cases} \qquad (9)$$

where θ is a threshold. The relation in an instance is predicted as the class Other if the score s_c is less than θ for every class c. Otherwise, we choose the class with the highest score as the prediction.

3 Experiments and Results

3.1 Data Preparation

We use both the Chinese and English corpus of ACE 2005 to evaluate our proposed approach. Only positive instances have been annotated in the dataset. To extract negative instances, we need to enumerate every entity pair in a sentence.

We consider two approaches: one considers the direction of relation while the other does not. For the first approach, we extract only one instance for any pair of entities e_1, e_2 in a sentence x regardless of direction. Those instances that have been annotated, regardless of direction, are positive instances, and the rest are negative instances. A trained model only needs to determine whether an entity pair has relation. For the second approach, we extract two instances for any pair of entities in a sentence, with the two entities in different orders. Since at most one of the two instances has been annotated to be positive instances, we treat the other one and those neither of which are annotated to be negative instances. A trained model will additionally need to identify head entity and tail entity in a relation, which is considerably harder.

After extracting negative instances, we further filtered out those instances whose entity type combination has never appeared in a relation mention. Then we added the remaining negative instances to the positive instances to complete data preparation.

We adopted the first approach to extract negative instances from the English corpus of ACE 2005, and ended up with 71,895 total instances after filtering, among which 64,989 are negative instances. The positive/negative instance ratio is about 1:9.4.

We adopted the second approach to extract negative instances from the Chinese corpus of ACE 2005, and ended up with 215,117 total instances after filtering, among which 205,800 of them are negative instances. The positive/negative instance ratio is about 1:20.

3.2 Experiment Settings

3.2.1 Embeddings

In our approach, we use three kinds of embeddings, namely word embeddings, position embeddings and BIO tag embeddings. They are all randomly initialized, and are adjusted during training. The dimensions of these three embeddings are 200, 50 and 50 respectively.

3.2.2 Hyper-parameters

The number of filters in the convolutional layer is 64, and the window size of filters ranges from 4 to 10. The fully connected layer to calculate class scores has 128 hidden units with a dropout rate of 0.2. The batch size is 256. The neural networks are trained using the RMSprop optimizer with the learning rate α set to 0.001.

As for the parameters in the pairwise ranking loss, for the English corpus, we set m^+ to 2.5, m^- to 0.5, γ to 2 and θ to 0; for the Chinese corpus, we set m^+ to 4.5, m^- to -0.5, γ to 2 and θ to 1. The cross-entropy loss and the pairwise ranking loss in multi-task learning are equally weighted.

3.3 Experiment Results

We use five-fold cross-validation to reduce the randomness in the experiment results. The precision (**P**), recall (**R**) and F1-score (**F1**) of the positive instances are used as evaluation metrics.

We compare several variants of our proposed models with the state-of-the-art models on the English and Chinese corpus of ACE 2005 respectively. Variants of our models are:

- Baseline: a model that uses CNN with filters of multiple window sizes and only performs the relation classification task using the pairwise ranking loss. The baseline model is motivated by dos Santos et al. (2015) and Nguyen and Grishman (2015).

- Baseline+Tag: baseline model with BIO tag embeddings.

- Baseline+MTL: baseline model that performs relation identification using cross-entropy loss in addition to relation classification.

- Baseline+MTL+Tag, baseline model that adopts both multi-tasking learning and BIO tag embeddings.

For the English corpus, we choose SPTree (Miwa and Bansal, 2016) and Walk-based Model (Christopoulou et al., 2018) for comparison. Since the data preparation is similar, we directly report the results from the original papers. The experiment results are summarized in Table 1.

For the Chinese corpus, we choose PCNN (Zeng et al., 2015) and Eatt-BiGRU (Qin et al., 2017) for comparison. We re-implemented these two models, and the experiment results are summarized in Table 2.

Model	P%	R%	F1%
SPTree	70.1	61.2	65.3
Walk-based	69.7	59.5	64.2
Baseline	58.8	57.3	57.2
Baseline+Tag	61.3	76.7	**67.4**
Baseline+MTL	63.8	56.1	59.5
Baseline+MTL+Tag	66.5	71.8	**68.9**

Table 1: Comparison between our model and the state-of-the-art models using ACE 2005 English corpus. F1-scores higher than the state-of-the-art are in bold.

Model	P%	R%	F1%
PCNN	54.4	42.1	46.1
Eatt-BiGRU	57.8	49.7	52.0
Baseline	48.5	57.1	51.7
Baseline+Tag	61.8	62.7	**61.4**
Baseline+MTL	56.7	52.9	**53.8**
Baseline+MTL+Tag	61.3	65.8	**62.9**

Table 2: Comparison between our model and the state-of-the-art models using ACE 2005 Chinese corpus. F1-scores higher than the state-of-the-art are in bold.

From Table 1 and Table 2, we can see:

1. Both BIO tag embeddings and multi-task learning can improve the performance of the baseline model.

2. Baseline+Tag can outperform the state-of-the-art models on both the Chinese and English corpus. Compared to the baseline model, BIO tag embeddings lead to an absolute increase of about 10% in F1-score, which indicates that BIO tag embeddings are very effective.

3. Multi-task learning can yield further improvement in addition to BIO tag embeddings: Baseline+MTL+Tag achieves the highest F1-score on both corpora.

3.4 Analysis

3.4.1 Effectiveness of BIO Tag Embeddings

To further investigate the effectiveness of BIO tag embeddings, we incorporated these embeddings into PCNN (Zeng et al., 2015) and Eatt-BiGRU (Qin et al., 2017) to form two new models: PCNN+Tag and East-BiGRU+Tag, and evaluated their performance using the Chinese corpus of ACE 2005. The results are summarized in Table 3.

Model	P%	R%	F1%
PCNN+Tag	74.3	50.4	58.2
Eatt-BiGRU+Tag	67.8	56.4	61.1

Table 3: Evaluation of state-of-the-art models with BIO Tag embeddings using ACE 2005 Chinese corpus.

Compare Table 3 with Table 2, and we can see that thanks to BIO tag embeddings, the F1-score of PCNN increases from 46.1% to 58.2%, while the F1-score of Eatt-BiGRU increases from 52.0% to 61.1%. Such significant improvement is consistent with that on the baseline model and further attests to the effectiveness of BIO tag embeddings. We believe that BIO tag embeddings could be used as a general part of character/word representation for other models and potentially other tasks as well.

3.4.2 Effect of Positive/Negative Instance Ratio

To see how our approach would perform as the degree of data imbalance varies, we used the same random seed to sample negative instances extracted from the Chinese corpus of ACE 2005 to add to the positive instances with different negative/positive instance ratios of 1:0.5, 1:1, 1:5, 1:10 and 1:15. Then we trained and evaluated two models: Baseline and Baseline+MTL+Tag. The results are shown in Figure 3.

As shown in Figure 3, the performance drops for both models in terms of F1-score as the positive/negative instance ratio decreases. Yet, as the data become more imbalanced, the gap between the performances of Baseline+MTL+Tag and Baseline widens. This indicates that our proposed approach is more useful when the data is

Model	RI	Loss Function in RC	P%	R%	F1%
Baseline+Tag	×	Ranking Loss	61.8	62.7	61.4
Baseline+Tag	×	Cross-entropy Loss	67.7	57.8	61.5
Baseline+Tag	×	Cross-entropy Loss + Ranking Loss	63.2	62.1	61.7
Baseline+MTL+Tag	✓	Ranking Loss	61.3	65.8	62.9
Baseline+MTL+Tag	✓	Cross-entropy Loss	61.6	62.0	62.0

Table 4: Evaluating the effect of the loss function used in relation classification w/o multi-tasking using ACE 2005 Chinese corpus. RC stands for relation classification and RI stands for relation identification.

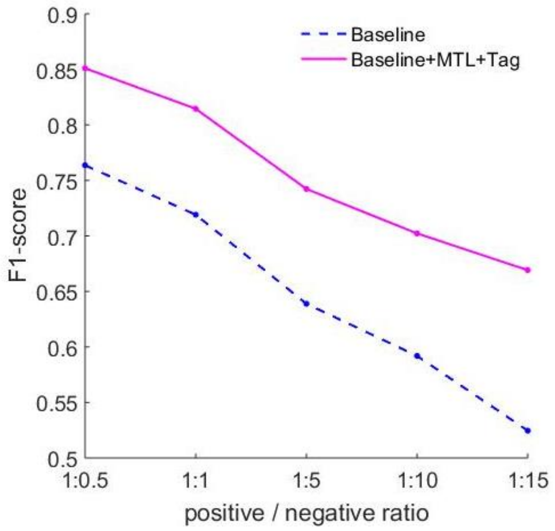


Figure 3: Effect of positive/negative instance ratio on F1-score.

more imbalanced, though it performs better than the baseline regardless of the positive/negative instance ratio.

3.4.3 Effect of Loss Function w/o Multi-tasking

Recall that in the multi-task architecture that we have proposed, we use the pairwise ranking loss for the multiple classification task of relation classification and use cross-entropy loss for the binary classification task of relation identification.

We can, however, use cross-entropy in relation classification as well. To see how the choice of loss function affects performance in different scenarios, we switched ranking loss to cross-entropy loss or simply added cross-entropy loss in the relation classification task, and evaluated the Baseline+Tag model w/o multi-task learning, using the Chinese corpus of ACE 2005. The results are summarized in Table 4, from which we can see:

1. When doing a single task of relation classification, the model has higher precision and lower recall with cross-entropy loss, but has lower precision and higher recall with ranking loss; the F1-scores do not differ much. This suggests that for doing relation classification only, the choice of loss function seems not to matter too much.

2. Multi-task learning helps, regardless of the loss function used in relation classification.

3. When we use cross-entropy loss and ranking loss at the same time for relation classification, without multi-tasking, the F1-score only increases slightly from 61.4% to 61.7%. But when cross-entropy is applied to another related task—relation identification, with multi-tasking, the F1-score increases from 61.4% to 62.9% with an absolute increase of 1.5%. This suggests that the effectiveness of our multi-task architecture mostly comes from the introduction of relation identification, and this binary classification task does help with the data imbalance problem, corroborating our motivation stated in Section 1.

4. In the same multi-tasking scenario, using ranking loss in relation classification is better than using cross-entropy loss (62.9% v.s. 62.0%), with an absolute increase of 0.9% in F1-score. Note that cross-entropy loss is already used in relation identification. This suggests that the diversity that comes with ranking loss can improve performance.

4 Related work

Liu et al. (2013) were the first to adopt deep learning for relation extraction. They proposed to use a CNN to learn features automatically without using handcraft features. Zeng et al. (2014) also employed CNN to encode the sentence, using additional lexical features to word embeddings. Their biggest contribution is the introduction of position embeddings. Zeng et al. (2015) proposed a model named Piecewise Convolutional Neural Networks (PCNN) in which each convolutional filter p_i is divided into three segments (p_{i1}, p_{i2}, p_{i3}) by head and tail entities, and the max-pooling operation is applied to these three segments separately. dos Santos et al. (2015) also used CNN but proposed a new pairwise ranking loss function to reduce the impact of negative instances. Lin et al. (2016) used CNN with a sentence-level attention mechanism over multiple instances to reduce noise in labels.

RNN is also widely used in relation extraction. Miwa and Bansal (2016) used LSTM and tree structures for relation extraction task. Their model is composed of three parts: an embedding layer to encode the input sentence, a sequence layer to identify whether a word is an entity or not, and a dependency layer for relation extraction. Zhou et al. (2016) used BiLSTM and attention mechanism to improve the model's performance. She et al. (2018) proposed a novel Hierarchical attention-based Bidirectional Gated recurrent neural network (HBGD) integrated with entity descriptions to mitigate the problem of having wrong labels and enable the model to capture the most important semantic information.

Entity background knowledge also contains important information for relation extraction. To capture such information, Ji et al. (2017) and She et al. (2018) extracted entity descriptions from Freebase and Wikipedia and used an encoder to extract features from these descriptions. He et al. (2018) used a dependency tree to represent the context of entities and transformed the tree into entity context embedding using tree-based GRU.

Unlike most existing works which only consider a single entity pair in a sentence, Christopoulou et al. (2018) considered multiple entity pairs in a sentence simultaneously and proposed a novel walk-based model to capture the interaction pattern among the entity pairs. Su et al. (2018) pointed out that the global statistics of relations between entity pairs are also useful, and proposed to construct a relation graph and learn relation embeddings to improve the performance of relation extraction.

Several studies are motivated to mitigate the effect of wrong labels (Lin et al., 2016; She et al., 2018; Qin et al., 2018), and Li and Ji (2014) proposed to jointly extract entity mentions and relations. This is not the focus of our paper.

5 Conclusion

In this paper, we focus on the relation extraction task with an imbalanced corpus. To mitigate the problem of having too many negative instances, we propose a multi-task architecture which jointly trains a model to perform the relation identification task with cross-entropy loss and the relation classification task with ranking loss. Moreover, we introduce the embeddings of character-wise/word-wise BIO tag from the named entity recognition task to enrich the input representation. Experiment results on ACE 2005 Chinese and English corpus show that our proposed approach can successfully address the data imbalance problem and significantly improve the performance, outperforming the state-of-the-art models in terms of F1-score. Particularly, we find BIO tag embeddings very effective, which we believe could be used as a general part of character/word representation.

Acknowledgments

We would like to thank Handan Institute of Innovation, Peking University for their support of our work.

References

Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*, volume 71.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 724–731.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 152–160.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2018. A walk-based model on entity graphs for relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 81–88.

Aron Culotta and Jeffrey S. Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Asso-*

ciation for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain., pages 423–429.

Claudio Giuliano, Alberto Lavelli, Daniele Pighin, and Lorenza Romano. 2007. Fbk-irst: Kernel methods for semantic relation extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 141–144.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434.

Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, and Min Zhang. 2018. See: Syntax-aware entity embedding for neural relation extraction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 113–120.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 402–412.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2124–2133.

ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. 2013. Convolution neural network for relation extraction. In *International Conference on Advanced Data Mining and Applications*, pages 231–242.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Raymond J Mooney and Razvan C Bunescu. 2006. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 68–74.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1378–1387.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 697–704.

Pengda Qin, Weiran Xu, and Jun Guo. 2017. Designing an adaptive attention mechanism for relation classification. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 4356–4362.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. DSGAN: generative adversarial training for distant supervision relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 496–505.

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 626–634.

Heng She, Bin Wu, Bai Wang, and Renjun Chi. 2018. Distant supervision for relation extraction with hierarchical attention and entity descriptions. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Yu Su, Honglei Liu, Semih Yavuz anda Izzeddin Gur, Huan Sun, and Xifeng Yan. 2018. Global relation embedding for relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 820–830.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 521–529.

Le Sun and Xianpei Han. 2014. A feature-enriched tree kernel for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 61–67.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 288–295.

Guodong Zhou, Min Zhang, DongHong Ji, and Qiaoming Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.