Relation Extraction Based on Relation Label Constraints

Kehua Miao, Kaihong Lin, Wenxing Hong and Chaoyi Yuan

November 19, 2020

Relation extraction based on relation label constraints

Kehua Miao
Xiamen University
Automation
Xia men, China
zxkd@xmu.edu.cn

Kaihong Lin
Xiamen University
Automation
Xia men, China
2455633398@qq.com

Wenxing Hong
Xiamen University
Automation
Xia men China
hwx@xmu.edu.cn

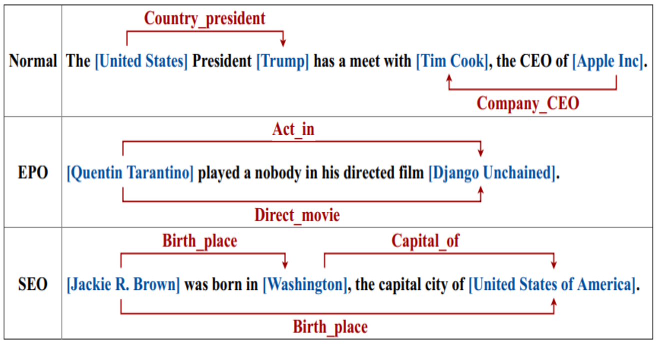
Chaoyi Yuan
North China Institute of\
Computing Technology
Bei jing, China
247133278@qq.com

Knowledge graphs have a significant role in promoting natural language processing tasks, and they have received substantial attention. The relation extractor is a key step in the construction of a knowledge graph, so it is important to improve its performance. However, previous works are mainly based on the pipeline method, which rarely address the problem of overlapping triplets. In addition, the literature does not consider models in which the correlation between relation pairs is addressed, which limits their accuracy. In this paper, we propose a new model called Relation Extraction Based On Relation Label Constraints(RRC) that is based on relation matrix constraints. The subject is extracted in our model in the first step; then, the relation and object are extracted based on the subject information. Each relation is regarded as a vector to assist in the extraction of the relation and object; the vector is used to consider the correlation between the relation vectors. This is used as a constraint to optimize the relation vector. Experiments on two public datasets, NYT and WebNLG, show that this method can perform well.

relation extractor, Knowledge graphs, Bert, relation matrix constraints

I. INTRODUCTION *(HEADING 1)*

Relation extraction is a task that extracts a triple from unstructured text, and these triples are the basic unit of building the large-scale knowledge graph. Knowledge graphs are applied to many natural language processing (NLP) tasks such as question answering, so it is very important to extract the triples from the text correctly. These triples are in the form of (subject, relation, object) or (s, r, o); they are referred to as relational triple. The relational triple in the sentence can be divided into three categories: Normal, EntityPairOverlap (EPO), and SingleEntityOverlap (SEO). These are illustrated in Figure. 1.


Normal The [United States] President [Trump] has a meet with [Tim Cook], the CEO of [Apple Inc].

EPO [Quentin Tarantino] played a nobody in his directed film [Django Unchained].

SEO [Jackie R. Brown] was born in [Washington], the capital city of [United States of America].

Figure 1: Examples of Normal, EntityPairOverlap(EPO) and SingleEntity- Overlap(SEO) overlapping patterns

As deep learning has achieved good results in many fields, it has also been used to solve NLP problems. In previous work in relation extraction, they took a pipeline approach [15], [20], [3]. It first recognizes all entities in a sentence and then performs relation classification for each entity pair. Such approach tends to suffer from the error propagation problem since errors in early stages cannot be corrected in large stages. To tackle this problem, subsequent works proposed joint learning of entities and relations. However, these models still have some limitations because they cannot solve the case where a sentence contains multiple triples, the overlapping entities between triples, and they do not take into account the correlation between the relation pairs.

Reference [17] proposed the first model to consider the overlapping problem in relational triple extraction. They introduced the categories for different overlapping patterns as shown in Figure 1 and used the sequenced-to- sequenced model to solve these problems. Moreover, they use reinforcement learning to further explore the impact of the order of extraction of triple on the result, which made progress on this task. [5] regard the sentence as relation graph and using graph neural network (GNN) to extract triples.

The previous model did not consider the correlation between relation pairs, Some relation pairs are strongly related, and some relation pairs are independent of each other. For example, the token corresponding to the entity that belongs to ‘founder' relationship is highly likely to belong to the ‘chairman' relationship, The 'chairman' and the 'lead actor' have a low probability of corresponding to these two relationships with the same token. In this paper, we propose a new approach to extract the triples, and define a relationship matrix to use a vector to represent a relation, after that, we calculate the correlation between the vectors and add it as a constraint to the loss function. The experiment show this model may well handle the various situations mentioned above and get better results. In this model, the extraction of triples is mainly divided into two steps. Firstly, our model extract subject using sequence tagging method, after that, we use the vector corresponding to extracted subject as the query and the hidden states of Bert output as key and value of the transformer block to learn a new hidden state containing the subject information. Finally, multiply the hidden state with the relation matrix to jointly extract the relationship and object.

This work has the following main contributions:

1.We introduce a new model for extracting triple from unstructured text.

2.add the correlation between the relationships as a constraint to the model.

The rest parts in this paper would be explained in following steps:

Firstly, the related works in relation extract field would be introduced.

Secondly, the model detail in this paper would be introduced.

Thirdly, some experiments that support this idea and model would be presented.

At last, we would conclude whole work in this paper.

II. RELATED WORK

The application of deep learning in direction of NLP is currently divided into two step: Firstly, training a distributed vector to represent a word, then used these vectors to do various downstream tasks. On this basis, various pretrained models were generated, such as ELMo [10], Bert [4]. In particular, the Bert model has reached SOTA in many down-stream tasks, such as question answering sentence classification. It is widely used that after this pretrained model came out, our work also uses the Bert model to encode the text.

Relation extraction is a sub-task of information extraction, and also a step to building a large-scale knowledge graph such as DBpedia [1], Freebase [2]. Early work in relational triples extraction took a pipeline approach. They extract relational triples in two steps: firstly running a named recognition(NER) on the input sentence to identify all entities and then running relation classification(RC) on pair of extracted entities. Using this pipeline method, the accuracy of the second step will be affected by preliminary errors. In order to ease these problems, many joint models such as [14]; [8]; [9]; [11], that aim to learn entities and relation jointly have been proposed. However, these models extract entities and relationships through parameter sharing and do not jointly decode the relationships and entities, resulting in the model not being able to learn the semantic information of the triple well. Instead of the model mention above, [19] implements joint decoding of entities and relations by regrading the task of relational triple extraction as an end- to-end sequence tagging problem without need of NER or RC.

None of the above models consider the overlapping of entity pair in the triples, [17] used sequence-to-sequence model to solve the problem of overlapping. Recently, [5] also studies the problem and propose a graph convolution networks(GCNs) based method. our model extracts subject firstly, and then using subject information, extract relation and object jointly method.

III. MODEL

Our model combines the semantic information of entities and relation to solve the overlapping of multiple triples. The model is mainly composed of the following modules: Bert Encoder Module, Subject tagger, Attention Between Subject and Hidden state, Object-Relation tagger. The overall framework of the model is shown in Fig.2:

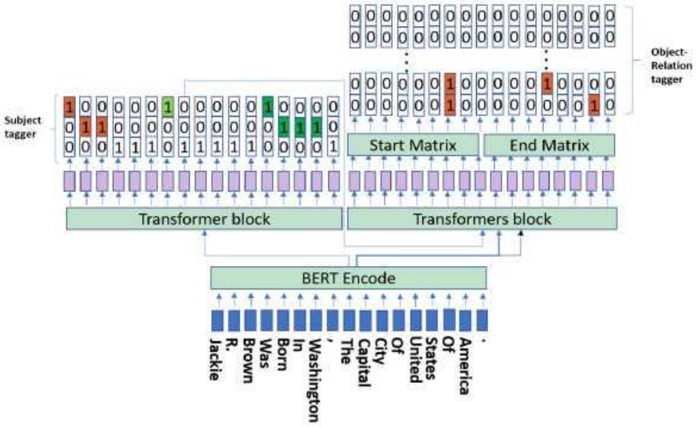


Figure 2: An overview of the proposed model framework structure.

A. Bert Encode Module

The BERT pre-training model includes Embedding module and 12 Trans- formers [13] Block modules. The Transformer module uses the multi-head attention to represent a word with a vector containing context information. The Trans(x) is defined as the function of Transformer module where x represents the input vector. We can get the following formula.

$$H_o = S * W_s + W_p \quad (1)$$

$$\text{Att} = \text{Attention}(W_q * H_{a-1}, W_k * H_{a-1}, W_v * H_{a-1}), a \in [1, N] \quad (2)$$

$$H_a = Trans(Att), a \in [1, N] \quad (3)$$

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\text{Q}, \frac{K^T}{\sqrt{d}}\right) * \text{V} \quad (4)$$

Where S is the matrix of one-hot vectors of token indices in the input sentence, W_s is the tokens embedding matrix, W_p is the positional embedding matrix, where p represents the position index in the input sequence, H is the hidden state vector, H_a representation the hidden state of input sentence at a-th layer and N is the number of Transformer blocks, W_v, W_k, W_q are the trainable parameters in the attention module. Since our input is a single sentence instead of a sentence pair in this task, we did not take segment embedding into account.

B. Subject Tagger

The output of the last layer of Bert would be utilized as the input of the subject tagger module. What's more, the sequence tagging process would be the vital method to extract subject. The BIO annotation method is employed by us to extract the subject, for it would be labeled B as it is in the beginning of an entity, that would be labeled I as it is inside the entity, that would be labeled O as it does not exist in the entity. From the Figure 2 you can see, Jackie R. Brown is a subject, so the label of Jackie is B, the label of R. and Brown is I. other word is labeled as O. Furthermore, we use the Softmax function to classify the tokens. The specific formula is as follows:

$$P_b = \frac{e^{Z_b}}{(e^{Z_b} + e^{Z_i} + e^{Z_o})} \qquad (5)$$

$$P_i = \frac{e^{Z_b}}{(e^{Z_b} + e^{Z_i} + e^{Z_o})} \qquad (6)$$

$$P_o = \frac{e^{Z_b}}{(e^{Z_b} + e^{Z_i} + e^{Z_o})} \qquad (7)$$

$$Z_j = W_j * x_i + B_j, \quad j \in [b, i, o] \qquad (8)$$

where the P_b represents the probability that this token belongs to the first word of the entity. P_i represents the probability that the token belongs to the internal word of the entity, and P_o represents the probability that the token does not belong to the entity word. We take the label with the largest probability value as the label of the token.x_i is the i-th vector of the H_n, W_j represents the trainable weight, and B_j is the bias.

The subject tagger optimizes the following likelihood function to identify the subject s given a sentence representation x:

$$P_e(o|s,x) = \prod_{j}^{L} P_i^j \qquad (9)$$

Where L is the length of the sentence. i in the [B,I,O], P_j^i is the probability of model output ,y_i is the label of the j-th token and y_j is equal to 1.

C. Attention Between Subject and Hidden State

As the formula below, V_s is assumed as one of the vector corresponding to the candidate subject, and V_p represents the position corresponding to the subject. In a general triplet, the two entities are not far apart, so the position information of the subject would contribute to the extraction of the object. Therefore, the position of the subject could be encoded as a new position vector V_p which is used to add V_s vector to get V vector. In all, the V vector and hidden state would be treat as the Query, Key and Value of the Trans Block module input without using multi-head attention to get the new hidden state. The specific formula is as follows:

$$V_p = W_p * S \qquad (10)$$

$$\mathrm{V} = V_p + V_s \qquad (11)$$

$$\mathrm{Att} = \mathrm{Attention}(W_v * V, W_k * H_{n-1}, W_v * H_{n-1}) \qquad (12)$$

$$H_{b_n} = Trans(Att) \qquad (13)$$

W_p represents the position embedding matrix, and S represents the position index of the subject in the sentence, and $H_{n-1}$is the hidden state of the N-1th layer of Bert, and Trans represents a Transformer block, The input of attention here is obtained by V, H_{n-1} through three fully connected layers

D. Object-Relation tagger

Start and end matrix are defined as two label_num * hidden_size relation matrices respectively, while label_num represents the number of categories of the relations, and hidden size represents the number of units output from the last layer of Bert. Through the multiplication of sentence matrix and relation matrix, calculating with sigmoid activation function, the score of each token vector and each relation vector could be obtained. The higher score is considered to be the token belonging to an object in the relation.

The higher scores are corresponding to the index of object in the relation matrix separately, for start matrix is used to extract the token at the beginning of the object, and the end matrix is used to extractor the token at the end of the object. When the value in the start matrix and end matrix is greater than the threshold value we set, it is set to 1, otherwise it is set to 0.The specific formula is as follows:

$$P_j^{start_i} = o(W_{start_i} * X_j + B_{start_i}) \qquad (14)$$

$$P_j^{end_i} = o(W_{end_i} * X_j + B_{end_i}) \qquad (15)$$

Where $P_j^{start_i}$ and $P_j^{end_i}$ represents the probability of identifying the j-th word in the sentence as start and end position of a object in the i-th relation respectively. O is sigmoid activation function.X_j is j-th vector in the sentence matrix and W_{start_i}, W_{end_i} is trainable variables.B_{start_i}, B_{end_i} is the bias.

The Object-Relation tagger optimizes the following likelihood function to identify the span of object o given the sentence representation x and a subjects:

$$P(o|s,x) = \prod_{t \subset s,e} \prod_{i}^{L} (P_i^t)^{I\{y_i^t=1\}} (1 - p_i^t)^{I\{y_i^t=0\}} \qquad (16)$$

Where y_i^s represents the the i-th token is the start of one object, and y_i^e represents the the i-th token is the end of one object,P_i^t is the model output of the i-th token.

Finally, the candidate subjects that are extracted by subject tagger would be passed through the above Attention Between Subject and Hidden State part and Object-Relation tagger one by one, then whole object and relation elements could be extracted for the triples.

E. Consstraints between relation pairs

We believe that the correlation between the relation pairs would affect the results of the relation extraction task. Assuming the co-occurrence frequency of the two relations are high, then we consider two relations are related so that the angles of the vectors corresponding to the two relations would be relatively lower. Otherwise, the co-occurrence frequency of two relations is low, the angle between the two vectors would be 90 degrees. Under this assumption, with the relation pairs vectors that are multiply between start and its compose matrix, as a constraint to the loss function. First, the co-occurrence frequency of the relations pairs should be counted on the training set to generate the adjacency matrix of the relations pairs. In order to generate an asymmetric matrix, the bilinear function is utilized to calculate the correlation between the relations matrix. And the adjacency matrix as a label for the correlation of the relation pairs vectors, the loss is set for optimize the relations pairs matrix. The specific formula is as follows:

$$r_{score} = W_R * W * W_R^T \tag{17}$$

$$\text{score} = \text{Activation}(r_{score}) \tag{18}$$

$$\text{loss} = \text{L}(\text{score}, \text{AM}) \tag{19}$$

Where W_R represents the relations matrix, W_R^T is the transpose of the relations matrix, and W is the trainable parameter, Activation is the activation function, and L is the loss function, AM is the adjacency matrix.

We use cross entropy as the loss function for extracting subject, object and relation. Ls and Lo to represent the loss of subject and object, plus the loss L_s of relational constraints. The final loss value optimized by the model is composed of three parts:L_s, L_o, and L_r.. The specific formula is as follows:

$$\text{L} = L_s + L_o + L_r \tag{20}$$

IV. EXPERIMENT

A. Experiment Setting

We use Bert as the encode module. The batch size is set to 32, maximum length of the sentence is 100, epoch is set to 10, learning rate is 3e-5, all dropout rates are set to 0.1, threshold is set to 0.4. We train the model by Adam stochastic gradient descent over shuffled mini-batches [7]. All our experimental results below are take the average from 5 run times.

B. Dataset

The model in this paper has been evaluated on two public data sets, NYT [12] and WebNLG [6]. The NYT dataset is generated by distance supervision [11]. It contains 1.18 million sentences and 24 defined relations. Each sentence in this dataset might contain multiple triples. The dataset is released by [18], which the training set contains 56195 samples and the validation set contains 5000 samples, and the test set contains 5000 samples. Table 1 shows some statistics of this dataset. WebNLG dataset was originally created for Natural Language Generation (NLG) tasks and adapted by [18] for relational triple extraction task which contains 246 predefined relation types.

Categoty	NYT		WebNLG	
	Train	Test	Train	Test
Normal	37013	3266	1596	246
EPO	9782	978	227	26
SEO	14735	1297	3406	457
ALL	56195	5000	5019	703

Table 1: Statistics of datasets. Note that a sentence can belong to both EPO class and SEO class.

C. Baseline and Evaluation Metrics

The following Table 2 is a comparison between our experimental results and other baselines. The basic model is evaluated on the NYT and WebNLG datasets. The experimental results from the table below show that the model in this paper is much better than other baseline models. In addition, we explored adding relational constraints to loss function on the NYT dataset.

model	NYT			WebNLG		
	Prec	Rec	F1	Prec	Rec	F1
NovaTagging [19]	62.4	31.7	42	52.5	19.3	28.3
CopyR MultiDecoder [18]	61	56.6	58.7	37.7	36.4	37.1
GraphRel [5]	63.9	60	61.9	44.7	41.1	42.9
CopyRRL [16]	72.8	69.4	71.1	60.9	61.1	61
RERLC	77.9	87.8	82.1	86	87.4	86.7

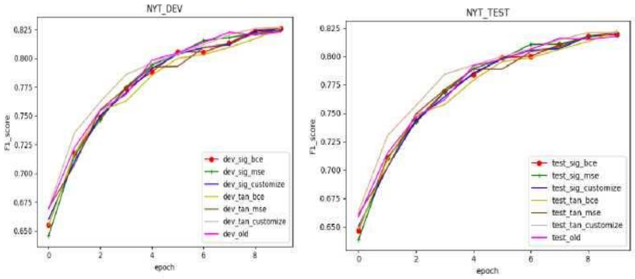
Table 2: Results of different methods on NTY and WebNLG datasets.

We explored several ways to add relation constraints to loss. It contains two activation functions sigmoid and tanh, and three loss functions: cross entropy loss function, hinge loss and mean square error function. The following table that show our experimental results on NYT datasets. It could be seen that our model has improved the F1 value compared to the loss without the relation constraints. The experimental results are shown in table 3 below

Model	NYT		
	Prec	Rec	F1
hinge loss	77.3	88.4	82.1
sig_bce	77.5	88.7	82.0
sig_mse	77.3	88.8	82.3
tanh_bce	77.6	88.2	82.2
tanh_mse	77.0	88.7	82.2
tanh_hinge_loss	78.1	88.3	82.3
None	78.0	87.8	82.1

Table 3: Results of different Loss function and Activation on NTY

We further explored the influence of different ways of relational constraints on the results. The following figure. 3 has two sub figures, fig. 33(a) is the dev result, and fig. 33(b) is the test result, shows our experimental results. For the left figure is the result of the validation set, and the right figure is the result of the test set. It can be seen that the use of the tanh activation function and the cross-entropy loss function improves the F1 value significantly at the beginning.



(a) The result of the F1 value of different model on the dev set

(b) The result of the F1 value of different model on the test set

Figure 3

V. Conclusion

In this paper, we propose a new model called Relation Extraction based on Relation Matrix Constrains (RERLC). This model regards the relation as the label of two entities, for the first of one extracts subject in the sentence, and then utilizes subject information to jointly extract relationship and object information. The model can well solve the problem of entity overlap between triples, and adding relational constraints to the loss function, which has improved the overall effect of the model. The model was verified the effectiveness of the model on the public data set of NYT WebNLG, and the experimental results show that our model can achieve better results.

References

[1] Soren Auer, Chris Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In Proceedings of the 6th International Semantic Web Conference (ISWC), volume 4825 of Lecture Notes in Computer Science, pages 722¬735. Springer, 2008.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structur¬ing human knowledge. In In SIGMOD Conference, pages 1247-1250, 2008.

[3] Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* pages 551-560, 2011.

[4] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language un¬derstanding. 2018.

[5] Tsu Jui Fu and Wei Yun Ma. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In ACL, 2019.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez- Beltrachini. Creating Training Corpora for NLG Micro-Planning. In *55th annual meeting of the Association for Computational Linguistics (ACL),* Vancouver, Canada, July 2017.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. Computer Ence, 2014.

[8] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In Meeting of the Association for Computational Linguistics, 2014.

[9] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation ex¬traction with table representation. In Conference on Empirical Methods in Natural Language Processing, 2014.

[10] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contex¬tualized word representations. 2018.

[11] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. Cotype: Joint extraction of typed entities and relations with knowledge bases. 2016.

[12] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling rela¬tions and their mentions without labeled text. In ECML/PKDD, 2010.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30,* pages 5998-6008. Curran Associates, Inc., 2017.

[14] Xiaofeng Yu and Wai Lam. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *International Conference on Coling*, 2012.

[15] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.

[16] Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* pages 367-377, Hong Kong, China, November 2019. Association for Computational Linguistics.

[17] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*

[18] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 2017.

[19] GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05),* pages 427-434, 2005.