IOP Conf. Series: Journal of Physics: Conf. Series **1487** (2020) 012009 doi:10.1088/1742-6596/1487/1/012009

A Joint Learning Information Extraction Method Based on an Effective Inference Structure

Shaopeng Ma[1,a] and Xiong Chen[1,b*]

[1]Department of electronic engineering, Fudan University, Shanghai 200433, China

[a] spma13@fudan.edu.cn; [b,*] chenxiong@fudan.edu.cn

Abstract. Over the past few years, natural language processing is getting much attraction from more scholars and institutions. Knowledge graph has been regarded as a crucial role in pushing natural language understanding forward. The task of information extraction is the first step to build a large-scale knowledge graph, which means to identify information from the natural language text and extract it in the form of entity and relation triplets. Some joint learning method have been proposed in this domain recently. In this paper, we inherit the idea of joint learning, use a simple, light-weight but effective structure to solve this task and compare our method with some recent algorithms on the benchmark dataset NYT and WebNLG. Results show that our method can get an improvement in F1 score.

1. Introduction

Over the past few years, theoretical and empirical researches of machine learning especially in deep learning have been in rapid development. As an important branch, natural language processing is getting much attraction from more scholars and institutions. Some front exploration and mature methods have already permeated in a few aspects in people's daily lives, such as machine translation, man-machine dialogue and search engine. Nevertheless, even in these much successful application scenarios, existing methods is not perfectly satisfied. In recent years, knowledge graph has been regarded as a crucial role in pushing natural language understanding forward. Benefiting from the appearance of large-scale knowledge graph, some downstream tasks have obtained substantial progress, such as question answering[1] and intelligent recommendation system[2].

The first step before enjoy the knowledge graph's bonuses is to construct a high-quality knowledge base, in which structured data being stored and managed. Knowledge here can be interpreted as a set of triplets in the form of <entity a, relation p, entity b>, which means there is a relation p between a and b. The task of information extraction is to identify information from the natural language text and extract it in the form of entity and relation triplets.

Generally, there are two different learning frameworks for this task, pipeline learning and joint learning respectively. Intuitively we can solve this problem by solving sub-tasks named entity recognition[3] and relation classification[4] respectively and subsequently. However, it is hard to neglect the inherent defect of upwards-magnifying errors through the tasks pipeline. In addition, dividing into independent tasks means paying few attentions to the important and actual correlation between them. If we don't grudge time to classify relations between each pair among all entities, this method maybe shows higher recall rate.

Given problems mentioned above of pipeline learning, scholars gradually shift more interest to joint learning[5]-[6] in recent years. Intuitively people extract information in sentences taking full advantage of mutual information shared between these two tasks. Its most noticeable feature is to recognize named entities and identify relations among them simultaneously by a single model, which avoid the upwards-magnifying errors born with pipeline learning. And many researches[5] show that joint learning can improve the performance in both tasks. The difficulty is how to design proper architecture to combine two sub-tasks. Given some previously widely referenced datasets, most existing joint learning methods only take account extract non-overlapping relations in one sentence, as shown in Table 1. So, it is hard to achieve the requirement of practice application.

Table 1. Examples of relation-level overlapping and entity-level overlapping from dataset NYT[5] and WebNLG[7] respectively.

	Sentence	Relations
Relation-level over-lapping	Google 's plans will be laid out by one of its two founders, Larry Page.	‘Google’, ‘founder’, ‘Larry Page’ ‘Google’, ‘shareholder’, ‘Larry Page’
	Jusuf Kalla is a leader in Indonesia where they speak Indonesian and eat Bakso.	‘Bakso’, ‘region’, ‘Indonesia’ ‘Bakso’, ‘conutry, ‘Indonesia’
Entity-level over-lapping	Mr. Hatcher was born in Ada, Oklahoma and was brought up in Tulsa.	‘Oklahoma’, ‘contains’, ‘Ada’ ‘Oklahoma’, ‘contains’, ‘Tusla’
	A.S. Roma play in Serie A and their stadium is in Rome.	‘A.S. Roma’, ‘league’, ‘Serie A’ ‘A.S. Roma’, ‘ground’, ‘Roma’

In this paper, we inherit the idea of joint learning, using a light-weight cascading gated convolutional neural network as encoder to process the syntactic and semantic information from raw texts. Then, we use a simple but effective pointer decoder network to transform the first entity extraction task into a tagging prediction problem. Finally, we use another pointer decoder to extract the corresponding entity of certain relation also in the form of a tagging prediction problem. Overall, this method is an end-to-end model, which is easy to train and effective in practical application. We compare our method with some recent algorithms on the benchmark dataset NYT and WebNLG.

2. Related work

2.1. Joint learning

Joint learning methods can be generally divided into parameter sharing and tagging schema. Zheng et al. [8] proposed one architecture, hybrid neural network of one LSTM and one CNN for named entity recognition and relation classification respectively, based on a shared feature extraction layers consisted of word embedding and Bi-LSTM layers. Novel-tagging-scheme is another influential joint learning method which doesn’t rely on feature engineering. This method embeds the relation classification task into traditional name entity recognition task by applying their well-designed tagging schema. The difference between lies in the combination of relation types and “BIES” (Begin, Inside, End, Single) signs. One obvious defect is the incapability of entity overlapping cases given this tagging schema. Zeng et al. [9] introduced their copy mechanism to extract overlapping entities. They transform the task into a sequence-to-sequence learning problem, and use multi-decoders to solve entities overlapping.

2.2. Neural network

In the domain of text generation, especially for the text summarization generation task, the pointer network [10] is widely used to extract target sentences from whole texts as a proper summarization. This method can be regarded as a sentence or phrase grained entities extraction, which can be referenced to this word grained entities extraction. With the rapid advancing of theoretical and

empirical researches of deep learning, hundreds of neural network architecture appear in natural language processing domain. Now the CNN architecture generally substitutes key roles of tradition RNN architecture and fully connected networks, in the consideration of its faster inference speed and light-weight in parameter size. After introducing of the dilation in CNN, the defects of weak capability of long-term dependency by effectively enlarging the respective filed. The recent gated linear dilated residual network proposed by Wu et al. [11] performs well in reading comprehension field.

Our method combines advancements of these mentioned researches, the parameter shared encoding layers, the tagging schema, the multi-entities processing method, an encoder with multi-dilated-CNN layers, and a pointer-network-like design in the extraction module.

3. Method

3.1. Structure Design

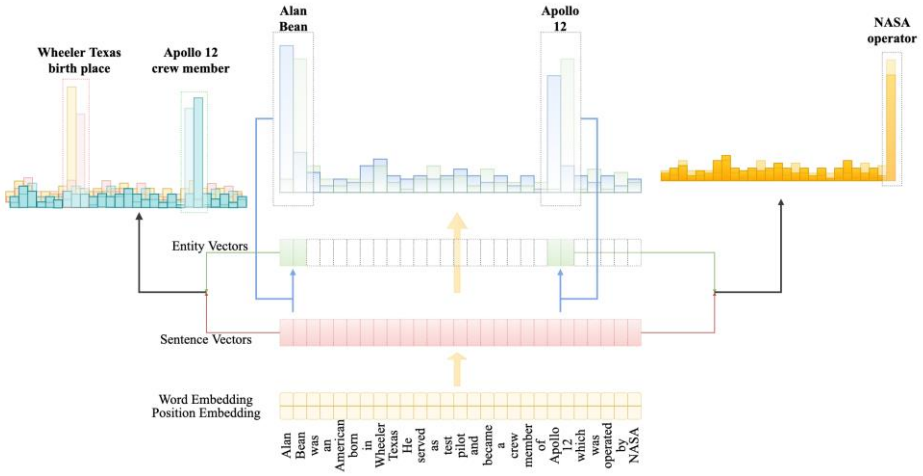

Figure 1. Illustration of our method

Figure 1 shows the overall framework of our method and its inference process of one case from dataset WebNLG, “Alan Bean was an American born in Wheeler, Texas. He served as a test pilot and became a crew member of Apollo 12, which was operated by NASA”. The middle histogram is the prediction of the start and end of entity (blue and green respectively). Only positions exceeding the threshold can be seen as the end of entities, and in this case, they are Alan Bean and Apollo 12. Then we extract corresponding vectors from whole sentence according to their position, feed into an attention layer to get entity vectors and concatenate them to sentence vectors respectively. Bilateral histograms are result of prediction of another entity and their relation. Different color represents different relations and deep and light color represents the start and end position respectively as mentioned before. The final results of this case are (Alan Bean, birth place, Wheeler Texas), (Alan Bean, crew member, Apollo 12) and (Apollo 12, operator, NASA).

3.2. Encoder

We define a sentence with t words, $s = [w_1, w_2, \cdots, w_t]$ where w_i represents i-th word. Like most of NLP tasks, we convert words into vectors by an embedding matrix in the embedding layer and get a vector-formed sentence, $s = [e_1, e_2, \cdots, e_t]$ where $e_i \in R^d$ represents i-th d-dimensional vector. To help CNN handle location-dependent tasks, we need combining manual designed position information to word vectors, as shown in equation 1, where p represents position in sentence, i represents dimension in vectors and d represent dimension of vectors.

$$PositionEncodering_{2i}(p) = sin(p/10000^{2i/d})$$
$$PositionEncodering_{2i+1}(p) = cos(p/10000^{2i/d}) \quad (1)$$

Then we use multiple gated convolutional network with dilation layers and self-attention layer as encoder.

$$(kernel * s)_{t,c} = \sum_{i=-l}^{l} \sum_{j} k_{i,j} \cdot e_{t-d \cdot i,j} \quad (2)$$

A convolution kernel of size $2l+1$ and dilation d can encode one sentence as shown in equation 2, where t and i represent positions in sentence, c and j represent channels.

$$s = s \cdot (1 - \sigma(k_1 * s)) + (k_2 * s) \cdot \sigma(k_1 * s) \quad (3)$$

A gated convolution layer can encode one sentences one sentence as shown in equation 3, where σ represents sigmoid function and k_1 and k_2 represent two different convolution kernels, k_1 playing the role like a gate.

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

A self-attention layer can weight vectors in matrix V, by the softmax function depending on the similarity between corresponding vectors pairs in matrix Q and K respectively, as shown in equation 4. Here matrix Q, K and V is outputs of three different non-linear layers of the same sentence vector series and scalar d is alse dimension of vectors. Self-attention is generally considered as the layer to enhance relations between any two correlated vector pairs.

3.3. Prediction module

As mentioned before, firstly, we use a pointer-like architecture to extract all candidate entities from sentences, which means we can regard the entity recognition as two binary classification problems. We need to predict whether the i-th word can be end of one entity, which is simpler than the prediction of BIES tagging. Its loss function can be defined by equation 5, where f_b and f_e are the final fully-connected neural network activated by sigmoid function (their function is to predict whether position i can be the begin or end position of entities respectively), and I represents the indicator function (only if right-bottom condition is satisfying can it output true value). This loss function is combination of two cross entropy along with whole sentence series.

$$loss_1 = \sum_i [\log(f_b(e_i)) \cdot I_{y_{i,b}=1}(y_i) + (1 - \log(f_b(e_i))) \cdot I_{y_{i,b}\neq 1}(y_i)] + \sum_i [\log(f_e(e_i)) \cdot I_{y_{i,b}=1}(y_i) + (1 - \log(f_e(e_i))) \cdot I_{y_{i,b}\neq 1}(y_i)] \quad (5)$$

Then we need to extract corresponding entity and relation based on word vectors and certain entities. The second task's loss function can be defined like equation 6, where r represents one relation, $f_{b,r}$ and $f_{e,r}$ is the function to predict whether i-th word can be the boundary of certain entity that has relation r with entity$_1$. In another word, we need to train $2r$ classifiers and predict $2rt$ times to traversal all relations and all positions to make sure that no potential candidate pairs are ignored. In this sense, our method can handle the entity overlapping problem.

$$loss_2 = \sum_r \{\sum_i [\log(f_{b,r}(e_i, e_1)) \cdot I_{y_{i,b,r}=1}(y_i) + (1 - \log(f_{b,r}(e_i, e_1))) \cdot I_{y_{i,b,r}\neq 1}(y_i)] + \sum_i [\log(f_{e,r}(e_i, e_1)) \cdot I_{y_{i,b,r}=1}(y_i) + (1 - \log(f_{e,r}(e_i, e_1))) \cdot I_{y_{i,b,r}\neq 1}(y_i)]\} \quad (6)$$

We use the sum of two losses as the whole loss function of this structure, which can share same parameters of encoder. We can consider that combining two losses prompt encoder to learn more common and more effective features for both tasks, which can also indicate that synergistic effect

between two steps works to solve the joint whole task. In this sense, joint learning has much more practical and theoretical significance.

4. Experiment

In this paper we consider two datasets to present out experimental results, NYT and WebNLG. NTY dataset has 236377 sentences with 25 relations in all. In our work we extracted 24 valid relations (except "None") and 66731 valid sentences (with valid relation pairs). As processed in the same way, WebNLG dataset has 22638 sentences with 246 relations proper for our research. We mix up the original train set and test set divided by their provider, and randomly select 10% as validation set, 10% as test set and the rest as train set.

Table 2. Comparison of experiments results of previous researches and our method in dataset NYT and WebNLG

Dataset	Model	Precision	Recall	F1
NYT	CoType	0.417	0.320	0.362
	NovelTagging	0.493	0.634	0.555
	Copy Mechanism	0.586	0.574	0.580
	Our Method	**0.702**	**0.579**	**0.635**
WebNLG	NovelTagging	0.525	0.193	0.283
	Copy Mechanism	0.377	0.364	0.371
	Our Method	**0.662**	**0.417**	**0.511**

We compares our method with mentioned CoType[5], NovelTagging[6] and Copy Mechanism[9], and our results indicate a better performance on dataset NYT and WebNLG for measurement index precision rate, recall rate and F1 score. As shown in Table 2, our method can get 0.702 precision, 0.597 recall and 0.635 F1 score in NYT, and get 0.752 precision, 0.466 recall and 0.575 F1 score in WebNLG. Our method can outperforms 5.5% and 14.0% improvements in F1 score respectively.

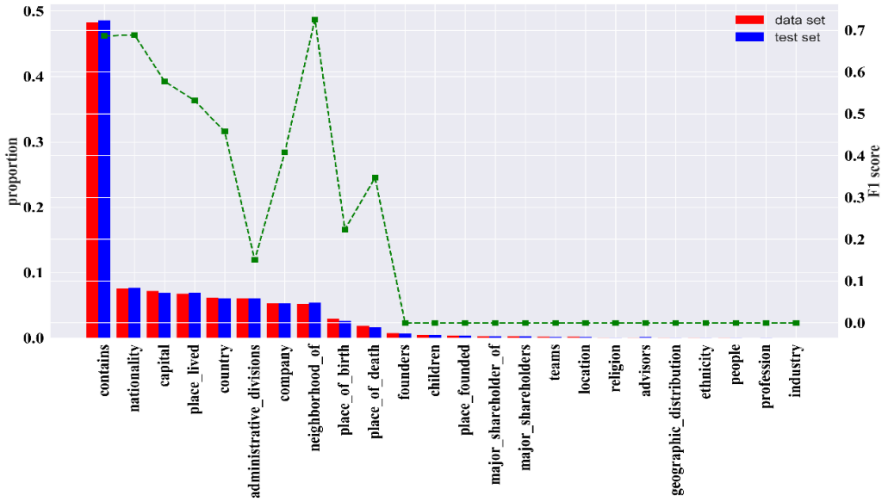

Figure 2. Relation distribution and F1 scores against different relations

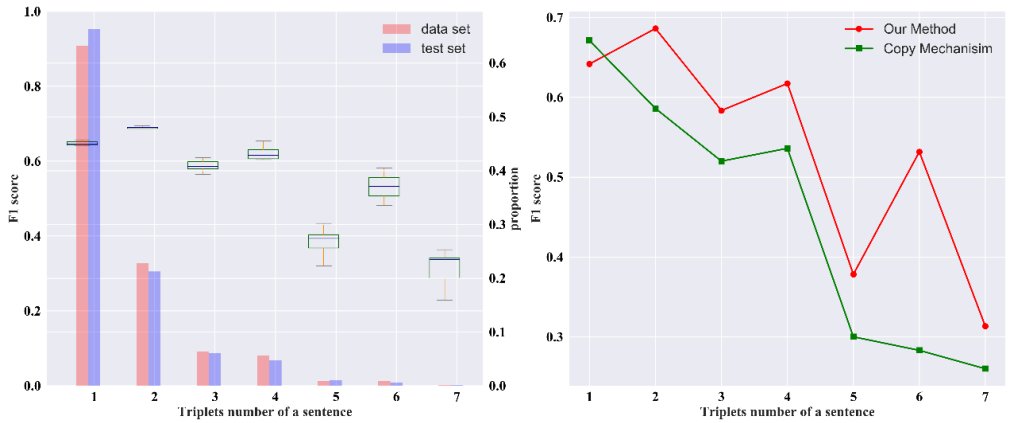

Figure 3. Triplet number distribution and F1 scores against different triplet number

In Figure 2, red and blue bar represent relation distribution in data set and test set respectively in dataset NYT, and green line represents F1 scores of different relations in test set (the zero value in the right space is because of the missing relation in test set). We can see that this dataset's relation distribution is much imbalanced, and the most common relation contains accounts for more than 45%. For 5 major relations our F1 score reach more than 0.5. Figure 3 shows the triplet number distribution in dataset NYT and our F1 scores of different cases. In the left subfigure, red and blue bar represent triplet number distribution in data set and test set respectively in dataset NYT, more that 90% sentences only have 1 or 2 relation triplets. And we use the boxplot to show the volatility of our method as the triplet number increases in sentence. The right subplot shows the comparison with Copy Mechanism. Our method performs better when triplet number is more than 1.

5. Conclusions and Future Work

In this paper, we focus on the task of information extraction from natural language text, and introduce an end-to-end joint learning method. Our method contains a simple but effective pointer network design and a parameter-shared light-weight CNN decoder. We make experiments and comparisons with some recent published researches on two widely-used public dataset. Results show that our method performs better F1 score than these baselines. But it is still not satisfied for practical application, the problem left is to continue improving the performance.

Acknowledgment

This work was supported by Shanghai Science and Technology Commission Project of China, No. 17DZ1201605.

References

[1] Aditya S, Yang Y and Baral C 2018 Explicit reasoning over end-to-end neural architectures for visual question answering *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* pp 629-637

[2] Wang H, Zhang F, Xie X and Guo M 2018 DKN: Deep knowledge-aware network for news recommendation *Proceedings of the 2018 World Wide Web Conference* pp 1835-1844

[3] Lample G, Ballesteros M, Subramanian S, Kawakami K and Dyer C 2016 Neural Architectures for Named Entity Recognition *Proceedings of NAACL-HLT* pp 260-270

[4] Wang L, Cao Z, De Melo G and Liu Z 2016 Relation classification via multi-level attention cnns *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (volume 1: long papers)* pp 1298-1307

[5] Ren X, Wu Z, He W, Qu, M, Voss, C R, Ji H and Han J 2017 Cotype: Joint extraction of typed entities and relations with knowledge bases *Proceedings of the 26th International Conference on World Wide Web* pp 1015-1024

[6] Zheng S, Wang F, Bao H, Hao Y, Zhou P and Xu B 2017 Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pp 1227-1236
[7] Gardent C, Shimorina A, Narayan S and Perez-Beltrachini L 2017 Creating Training Corpora for NLG Micro-Planners *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pp 179-188
[8] Zheng S, Hao Y, Lu D, Bao H, Xu J, Hao H and Xu B 2017 Joint entity and relation extraction based on a hybrid neural network *Neurocomputing*, *257*, 59-66
[9] Zeng X, Zeng D, He S, Liu K and Zhao J 2018 Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pp 506-514
[10]. See A, Liu P J and Manning C D. 2017 Get To The Point: Summarization with Pointer-Generator Networks *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pp 1073-1083
[11] Wu F, Lao N, Blitzer J, Yan, G and Weinberger K 2017 Fast reading comprehension with convnets *arXiv preprint arXiv:1711.04352*