DoughnutPRIS at TAC KBP 2016

Zuyi Bao, Yao Lu, Tian Tian, Dongyun Liang, Yinge Zhao, Zhichao Li, Pengda Qin, Weiran Xu
Pattern Recognition and Intelligence System Lab
Beijing University of Posts and Telecommunications
baozuyi@bupt.edu.cn

Abstract

This paper describes doughnutPRIS system for two tasks in TAC KBP 2016 track, Cold Start Slot Filling and Event Argument Extraction and Linking. Compared with the adopted systems last year, we continue to use end-to-end neural networks to tackle these information extraction problem; however, some latest architectures and tricks are integrated into the presented system. Moreover, we draw a lesson from the deficiencies of past, and make some specific adjustments in pre-processing and post-processing.

1 Introduction

Recently, deep learning techniques are widely used in various NLP tasks. Consequently, in TAC KBP 2015, we attempted to construct neural network systems for targeted tasks. Our works in this year are based on the experience of last year's system. We still participate in the Clod Start SF and EAL task. Notably, for EAL task, the scale of released corpus is much larger than that of TAC2015. Aimed at this change, we make the appropriate adjustments.

With respect to network structure, we adopt multi-layer Convolutional Neural Network(CNN) and Bidirectional Gated Recurrent Unit(GRU) (Cho et al., 2014) with attention mechanism (Bahdanau et al., 2014). Compared with the single layer CNN that we used in TAC2015, multi-layer structure can extract higher level feature; moreover, it is conducive to alleviate noise information in long sentences. Attention mechanism is a triumphant achievement in the field of deep learning and has widely applied in processing various NLP tasks. It is capable of automatically concentrating on qualitatively informative words for targeted task, which conforms to the concept of trigger word in relation and event extraction. The crucial information for determining relation and event mainly concentrate on the pattern that consists of trigger word.

In view of network structure, CNN performs better on recognizing consecutive pattern; however, RNN is good at mining long-distance discrete pattern. In order to combine there advantages, for each task, we build corresponding CNN-based and RNN-based systems. And the final results are selected by a predefined threshold.

The remainder of this paper is organized as follow. Section 2 describes some text processing details to generate normative sentence-level input. Section 3 and Section 4 respectively present the neural networks for Cold Start SF and EAL. Finally, in Section 5, we make a summary of our TAC2016 systems and point out the works that need to be strengthened in future.

2 Extract candidate sentences

DoughnutPRIS neural networks need sentence-level input, so the first work is to transform document-level plain texts to sentence-level neat texts that contain targeted query information. Figure 1 shows the operation details. It is similar to the process used in TAC2015. The only difference is that we augment the scale of external entity dictionary to 16909 items.

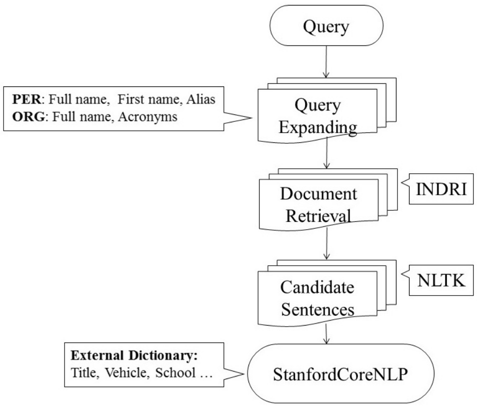

Figure 1: Candidate sentences extraction process

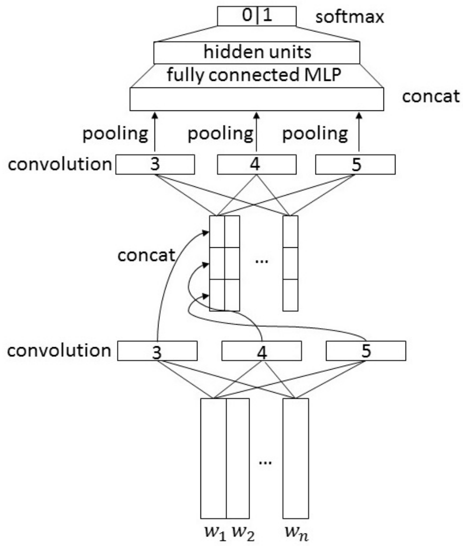

Figure 2: Multi-layer CNN architecture

3 Cold Start Slot Fiiling

We utilize two networks, multi-layer CNN (Zeng et al., 2014) and Bidirectional GRU (Xu et al., 2015), to jointly finish Cold Start SF task. The instances that both two networks have consistent judgements are naturally selected as final results. However, for the inconsistent instance, we simply chose the result with higher softmax probability. The official score for our SF system is presented in Table 4.

3.1 Muti-layer CNN

Figure 2 depicts the architecture of the multi-layer CNN. In order to extract more consecutive pattern information, we adopt three kinds of sliding windows, which represent 3-gram, 4-gram and 5-gram. After the first convolutional layer, we obtain the phrase-level features. Then, we exert the same convolution layer on these phrase-level features. Through this operation, we increase the length of consecutive patterns, which is beneficial to the relation expressions with long pattern.

Words in sentence are parametrized into word embeddings. The other important input is to point out the positions of query and candidate filler. Proved through the experiments that, combining Position Feature (PF) (Zeng et al., 2014) with Position Indicator (PI) (Zhang and Wang, 2015) achieves better performance than using any one of them. PF transforms the relative distances from context to query (or candidate filler) into distributed representations. PF of word that occurs on the left of query (or candidate filler) is encoded as the vector of negative number, otherwise positive number. These vectors are concatenated with the corresponding word embedding. PI adds four position indicators surrounding query and candidate filler. These indicators are assigned to distributed representations with the same dimension as word embedding.

3.2 Bidirectional GRU with attention

The conventional RNN exists the biased problem. To overcome this issue, the adaptive gating mechanism is adopted to alleviate this dilemma. Moreover, in comparison to LSTM, GRU contains less parameters to learn. So we select GRU to model sequence data.

Standard GRU is the unidirectional network, which means current hidden state only have access to the past context along input sequence in temporal order. However, relation classification depends on the overall information of sentence. Future words equally have impact on the semantic of past content. To solve this problem, we adopt a bidirectional architecture as in Figure 3 to summarize information for words from both directions. Standard GRU is regarded as the forward GRU. Backward GRU is used to process sequences in opposite temporal order. The corresponding hidden states are combined by concatenation operation. Then, these new hidden states are weighted by attention layer. The

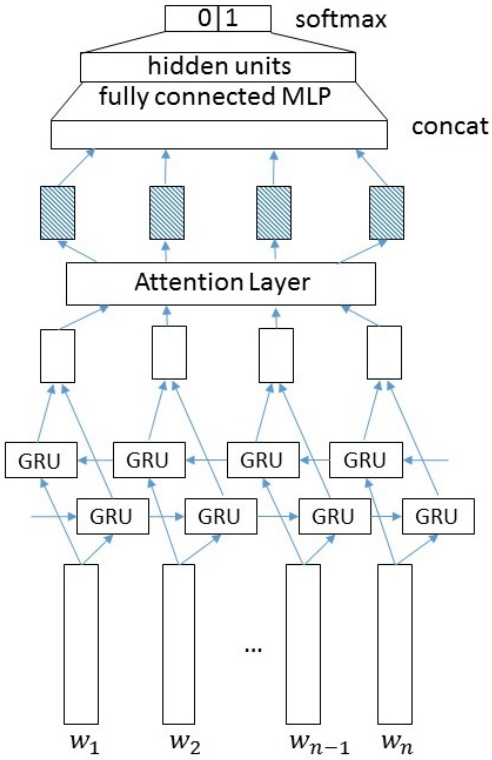

Figure 3: The architecture of Bidirectional GRU with attention

weights are calculated through the dot product between a random-initialized vector and new hidden states. This random-initialized vector is trained via system training.

As can be seen from Figure 2 and Figure 3, the final layer is a binary classifier. Before transmitted into network, the annotated input sentences are gave a predefined label according to Named Entity Recognition. The network only need to determine whether the input belongs to the predefined label or not. So, different relation types has different set of network parameters.

4 Event Argument Extraction

We divide EAL task into three parts, trigger identification (Nguyen and Grishman, 2015), argument extraction (Chen et al., 2015) and Realis detection. Detailed descriptions are presented in the ensuing subsections. Through observing the errors in TAC2015, we find that most of our submissions are incomplete. These standard answers are in phrase level; however, we only submit word-level answers. So,

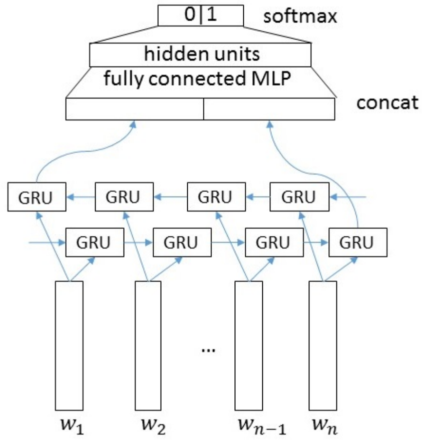

Figure 4: The architecture of Bidirectional GRU for EAL

in TAC2016, we add a chunking process as post-processing to compensate for this deficiency. Table 2, Table 3 and Table 1 give the official score of our EAL system.

4.1 Trigger identification

We use the same multi-layer CNN structure for trigger identification. Candidate trigger set is composed of the set selected from training set and the expanded set extracted from WordNet and FrameNet. The position information of candidate trigger in input sentence is encoded by PF and PI.

4.2 Argument extraction

Figure 4 presents the RNN-based network structure for argument extraction. In general, it is similar to network of SF task. However, attention mechanism is not adopted. Because, in input instance, trigger word is known to system. The addition of attention layer may be reduplicate. The other difference is that we only transmit the last hidden states of forward and backward GRU to the next layer. The 5-fold validation on training data shows that this framework can yield the same performance as Figure 3; moreover, it has better computational efficiency.

4.3 Realis detection

To solve this problem, we employ the network structure in Figure 2. Because there are only three labels (Actual, Generic and Other), we treat this subtask as multi-class problem. The other difference

System	TP	FP	FN	ArgP	ArgR	ArgF1	ArgScore	LinkScore
doughnutPRIS2	53.0	166.0	6450.0	24.2	0.8	1.6	0.6	0.1

Table 1: Official Score for doughnutPRIS2 submission

System	5%	50%	95%
doughnutPRIS2	0.4	0.6	0.7
Max	8.6	9.7	10.9
Rank_4	2.5	3.0	3.4

Table 2: Document Level Argument Summary Score of doughnutPRIS2 submission

System	5%	50%	95%
doughnutPRIS2	0.0	0.1	0.2
Max	7.8	8.7	9.6
Rank_4	1.3	1.6	2.0

Table 3: The Argument Score of doughnutPRIS2 submission

is that we substitute a pairwise ranking loss function (dos Santos et al., 2015) for the regular softmax, which makes it easy to reduce the impact of artificial classes.

4.4 Extract Phrase-level text

In order to expend the output of neural network, we utilize the analysis results of Stanford Parser[1]. As the parsing tree in Figure 5, we extract the first NP unit from bottom to top as candidate argument. However, not all situations are desirable, such as the NP unit in dotted box. Therefore, we design some rules to discard noise data.

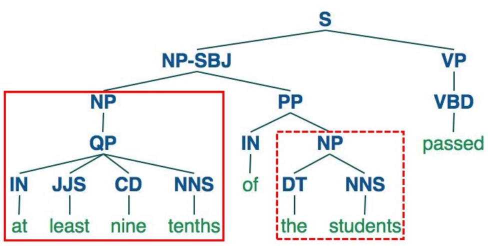

Figure 5: The architecture of Bidirectional GRU for EAL

5 Conclusion

This paper presents the doughnutPRIS system for Cold Start SF and EAL tasks in TAC2016. Compared with our last year's system, we continue to use neural network to tackle these problems; however, we adopt deeper network structure and make some specific modifications. There are still some deficiencies. We pay more attention to the improvement of network structure, but neglect the importance of text processing, such as coreference resolution. Moreover, some rule-based and feature-based methods are quite effective and can compensate the shortage of neural network. So, in future work, we will consider to integrate conventional methods with neural network. We are confident that it will improve our performance.

[1] http://stanfordnlp.github.io/CoreNLP/

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling 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*, volume 1, pages 167–176.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cıcero 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*, volume 1, pages 626–634.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation 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*, volume 2, pages 365–371.

QID/EC	RunID	Hop	Prec	Recall	F1
SF-ALL-Micro	doughnutPRIS_SF_ENG_5	0	0.1279	0.1036	0.1145
SF-ALL-Micro	doughnutPRIS_SF_ENG_5	1	0.0000	0.0000	0.0000
SF-ALL-Micro	doughnutPRIS_SF_ENG_5	ALL	0.1279	0.0687	0.0893
SF-ALL-Macro	doughnutPRIS_SF_ENG_5	0	0.1321	0.1525	0.1301
SF-ALL-Macro	doughnutPRIS_SF_ENG_5	1	0.0000	0.0000	0.0000
SF-ALL-Macro	doughnutPRIS_SF_ENG_5	ALL	0.0819	0.0946	0.0806
LDC-MAX-ALL-Micro	doughnutPRIS_SF_ENG_5	0	0.1293	0.0980	0.1115
LDC-MAX-ALL-Micro	doughnutPRIS_SF_ENG_5	1	0.0000	0.0000	0.0000
LDC-MAX-ALL-Micro	doughnutPRIS_SF_ENG_5	ALL	0.1293	0.0652	0.0867
LDC-MAX-ALL-Macro	doughnutPRIS_SF_ENG_5	0	0.1358	0.1539	0.1318
LDC-MAX-ALL-Macro	doughnutPRIS_SF_ENG_5	1	0.0000	0.0000	0.0000
LDC-MAX-ALL-Macro	doughnutPRIS_SF_ENG_5	ALL	0.0825	0.0934	0.0800
LDC-MEAN-ALL-Macro	doughnutPRIS_SF_ENG_5	0	0.1275	0.1456	0.1242
LDC-MEAN-ALL-Macro	doughnutPRIS_SF_ENG_5	1	0.0000	0.0000	0.0000
LDC-MEAN-ALL-Macro	doughnutPRIS_SF_ENG_5	ALL	0.0774	0.0884	0.0754

Table 4: Cold Start SF Official Score for our system

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (to appear)*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.