BERTatDE at SemEval-2020 Task 6: Extracting term-definition pairs in free text using pre-trained model

Huihui Zhang and **Feiliang Ren**
Computer science and Engineering, Northeastern University
Key Laboratory of Data Analytics and Optimization for Smart Industry
(Northeastern University),Ministry of Education, Shenyang, China
renfeiliang@mail.neu.edu.cn

Abstract

Definition extraction is an important task in Nature Language Processing, and it is used to identify the terms and definitions related to terms. The task contains sentence classification task (i.e., classify whether it contains definition) and sequence labeling task (i.e., find the boundary of terms and definitions). The paper describes our system BERTatDE in sentence classification task (subtask 1) and sequence labeling task (subtask 2) in the definition extraction (SemEval-2020 Task 6). We use BERT to solve the multi-domain problems including the uncertainty of term boundary that is, different areas have different ways to definite the domain related terms. We use BERT, BiLSTM and attention in subtask 1 and our best result achieved 79.71% in F1 and the eighteenth place in subtask 1. For the subtask 2, we use BERT, BiLSTM and CRF to sequence labeling, and achieve 40.73% in Macro-averaged F1.

1 Introduction

Definition extraction refers to extracting term-definition pairs from real texts. It is of great significance in technology and application development of information processing. Because it can be applied to many scenarios. For instance, it can be used to construct domain dictionary, knowledge graph and automatic question answering system, etc. And it also can improve the efficiency of search engine to a certain extent. As we all known, nature language is always complex, for example, polysemy is a pervasive phenomenon for some words. There are two complicate aspects.

First, term-definition pairs appears in different sentences. **Second**, there are lack of explicit definitions in some sentences. We can't find obvious sign of definable words, like *is*, *means*, *is defined as*, etc.

Considering these real situations, the DEFT corpus (Spala et al., 2019) collects some data that can reflect the reality. SemEval-2020 task 6 (Spala et al., 2020) and it is called by DeftEval for short. DeftEval sets up three subtasks on the DEFT corpus.

Subtask 1 is a sentence classification task. For a giving sentence, subtask 1 is used to classify whether it contains a definition. If the definition is included, the corresponding sentence is marked as '1'. **Subtask 2** is a sequence labeling task. For each words including punctuation in a sentence, subtask 2 is used to label them with BIO tags according to the corpus' tag specification. **Subtask 3** is a relation classification task. Given the tag sequence labels, subtask 3 is used to label the relations between each tags according to the corpus' relation specifications.

In the DeftEval, there are seven fields, the proportion of these fields are different. And different fields have different terms. This will bring some challenges such as **the boundary of terms is uncertain**. As each domain has different ways to describe the domain terms. Some terms appear less and may lead to the problem of out-of-vocabulary (OOV) .

BERT can capture polysemy and context sensitive embedding, which can produce more accurate vector representation to improve the performance of system. And BERT is equivalent to data enhancement, as it makes full use of a large amount of unsupervised data. So it can bring some semantic information to promote the performance of subtasks in DeftEval. Considering that CNN is often used in image and

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 690–696
Barcelona, Spain (Online), December 12, 2020.

speech recognition, it is not fully suitable for learning time series. LSTM can be well used in time series prediction. And compared with BiLSTM, the semantic information which captured by LSTM is incomplete. So we use BiLSTM. For subtask 1, we use BiLSTM and attention to further capture the semantic information of the context on the basis of BERT. For subtask 2, we use BiLSTM and CRF to sequence labeling on the basis of BERT.

2 Related Work

The traditional definition extraction technologies rely on manually defined features and rules, for example, (Westerhout,) used *linguistic and structural features* to extract definition. (Špela Vintar, 2012) used *Morphosyntactic patterns*, automatic terminology recognition and semantic tagging with wordnet senses are used to extract definition candidates from domain-specific corpora.

In recent years, a number of machine learning methods are applied in the definition extraction. For instance. (Przepiórkowski, 2008) used *Random Forest* that is used to identify definitions in Polish documents. (Ronzano, 2015) used *Weakly Supervised methods* that is used to extract textual definitions from naturally occurring text. A supervised approach is applied to Definition Extraction in which only syntactic features derived from dependency relations are used by (Espinosa-Anke and Saggion, 2014). A *joint model* (Veyseh et al., 2019) is applied to extract definition and its can keep syntactic connection and semantic consistency. Some Neural network methods are involved. For example, (Schockaert,) used neural architectures combining *Convolutional and Recurrent Neural Networks* and further enrich by incorporating linguistic information via syntactic dependencies.

3 System Overview

This section we will show the details about BERTatDE[1]. First, we will introduce about the pre-trained model BERT(Devlin et al., 2018). And next we will introduce another setups about the BERTatDE.

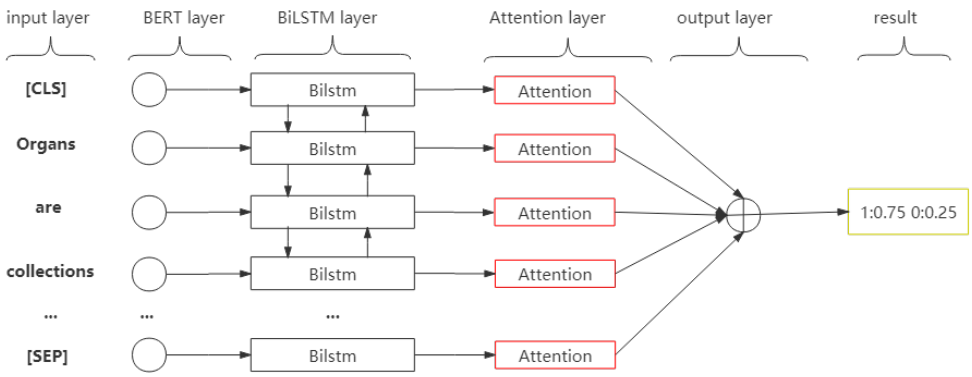


Figure 1: The structure of subtask 1 which contains five layers.

3.1 BERT

In this task, we need to extract the relevant definitions in the seven fields as Figure 3 (a) shows. On the one hand, employing traditional methods in these fields are time-consuming. On the other hand, pre-trained models have a outstanding performance in NLP fields and these models can bring some semantic information. Considering the above factors, we use pre-trained model. In this paper, we use BERT.

Transformer (Vaswani et al., 2017) is the main framework of BERT. Different from word2vec, BERT(Devlin et al., 2018) uses Masked LM and Next Sentence Prediction to capture representation at the word and sentence level respectively. And BERT makes full use of a large amount of unsupervised

[1]It is about the results of participant zhanghh2020 in subtask 1 and zhang-nlp subtask 2 of SemEval task6, and the result of subtask 2 is finished after official evaluation.

data, it can implicitly introduce linguistic knowledge into specific tasks. Meanwhile, BERT can introduce semantic information into embedding. It will help us to solve this multi-domain problem.

3.2 BERTatDE at Subtask 1

The input of BERTatDE of subtask 1 are sentences which separated by line break, and special marks '[CLS]' and '[SEP]' are added at the beginning and end of the sentence respectively. After processing the data (shows at 4.2) into standard input data as the Figure 1 input layer shows, we use BERT layer to get the word representation, and we use BiLSTM layer and attention layer to further gain semantic information, the architecture as Figure 1 shows.

LSTM (Hochreiter and Schmidhuber, 1997) its full name is Long Short-Term Memory, which can solve the long-term dependence of traditional RNN. The main idea is to introduce a gate mechanism, which can control the degree of historical information retained by each LSTM unit and remember the current input information, retain important features, and discard unimportant features.

However, LSTM don't take the backward information into consideration. To solve this problem, BiLSTM which combines forward LSTM with backward LSTM is proposed. After being encoded by BiLSTM, it can capture the context information in sentences. The output of BiLSTM is determined by the state of the hidden layer of these two LSTMs.

(Zhou et al., 2016) shows that since important information can appear at any position in the sentence. In order to capture the importance information in the sentence, we use attention layer. It can assign different weights to show that different words have diverse degrees of importance.

There are some ways to compute attention values. The method of calculating attention is proposed by (Zhou et al., 2016), we make use of these formulas to compute attention values.

$$M = tanh(H^T * \omega) \tag{1}$$

$$\alpha = softmax(M * \mu^T) \tag{2}$$

$$\gamma = H * \alpha^T \tag{3}$$

Among these formulas, H refers to the output of BiLSTM, α refers to the value of attention, γ refers to the output of the attention layer, μ and ω are hyper-parameters.

3.3 BERTatDE at Subtask 2

The input layer and BERT layer of subtask 2 are similar to subtask 1, we use BiLSTM layer to further obtain semantic features. And we employ CRF layer, (Huang et al., 2015) shows that it can use sentence level tag information.

As for subtask 2, it can be solved without using CRF. As the Figure 2 (a) shows the output of BiLSTM for each word is the label score. We can choose the label with the highest score for each word. This Figure don't consider special labels including '[CLS]' and '[SEP]', for the word 'Organisms', we choose the label corresponding to the highest score, that is 'B-Term'. The remaining tags are similar to 'Organisms'. However, the following situations may occur. It is likely that 'I-definition' appear after 'I-Secondary-definition'. Because 'I-Secondary-definition' refers to the definition which is based on another definitions, and the token is in the middle of the entity, 'I-definition' don't rely on another definitions. So there is no doubt that although each word selects the label with the highest score, the output results are not logical.

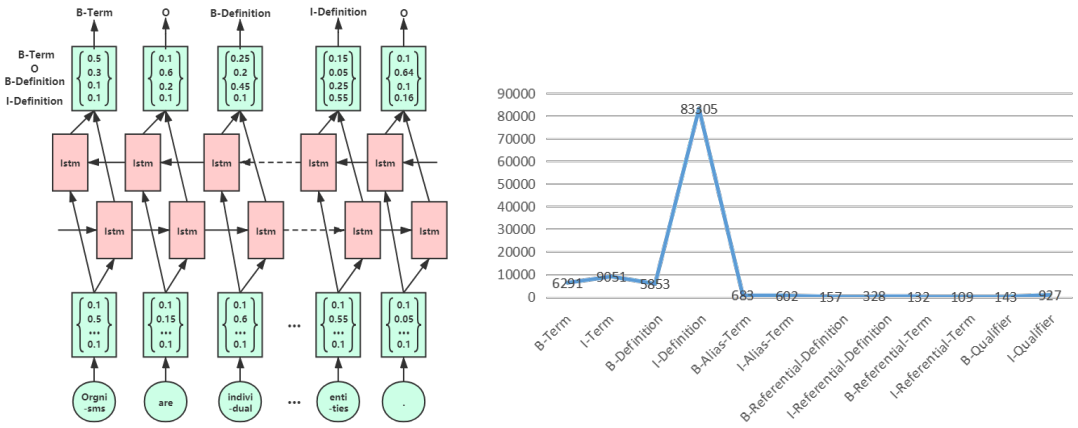
Under these circumstances, CRF (Sha and Pereira, 2003) can solve this problem. (Lample et al.,) shows that CRF will add come constraints to ensure the validity of the final prediction results, and invalid results will be reduced.

4 Experimental setup

4.1 Datasets

Datasets are available in official github[2]. The corpus is composed of English text, which contains seven domains including history, economic, government and so on. For the training data, the proportion of data

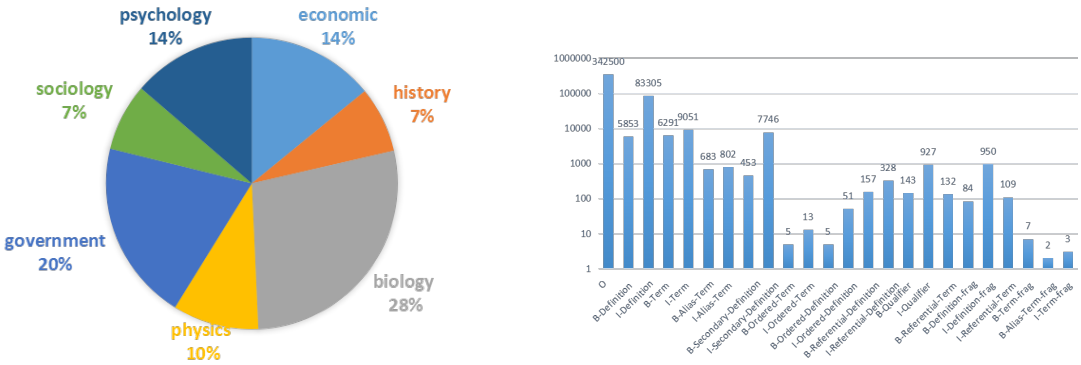
[2] https://github.com/adobe-research/deft_corpus



(a) (b)

Figure 2: (a):The affect of CRF in sequence labeling. (b):The distribution of evaluation tags in training set.

contained in each domain is shown in the Figure 3 (a). We can see the data in the biological field account for a larger proportion. The develop set its data distribution is similar to the training set.



(a) (b)

Figure 3: (a):The proportion of data in different fields in train set. (b):The proportion of each tags for train set.

Figure 3 (b) shows the proportion of 24 kinds of tags in the training set for subtask 2, including tags:O, B-Definition, I-Definition, B-Term, I-Term, etc. And there are connections between some tags, such as B-Definition, B-Secondary-Definition, which shows the sentence exist two kinds of definitions, the second definition is based on the first definition. We can see intuitively that some labels account for a large proportion such as 'O' about 74.52% (342500/459600), about 25.48% are another tags in the train set. In the develop set, the 'O' token about 74.578% (16402/ 21993), the rest of tags are approximately 25.422%.The Figure 3 (b) clearly shows that there is an imbalance between the tags.

4.2 Data Format

The standard data for subtask 1 is like the sentence "A variable is any part of the experiment that can vary or change during the experiment ." and corresponding label "1" .

As for subtask 2, we use corpus tags to label above sentences. The input data are series of words with its labels, such as "this" with its tag "O". When we deal with the data of subtask 2, we only use important corpus not all the corpus information.

4.3 Tools and libraries

In the system, we use PyTorch-Transformers which is a library of outstanding pre-trained models in Natural Language Processing (NLP). We employ *pytorch_transformers* to call the module of BERT. There are three modules including *BertModel*, *BertConfig*, *BertTokenizer*. And we use a *uncased* version of *Base* BERT. The module optimizer is *AdamW* which implemented in the module of PyTorch-Transformers[3]. When it comes to CRF, we use pytorch-crf[4] to call CRF modules. During the training process, the ReduceLROnPlateau is used to dynamically reduce the learning rate. The specific parameters about ReduceLROnPlateau are as follows:$mode =' max', factor = 0.5, min_lr = 1e-7, patience = 5, verbose = True, threshold = 0.0001, eps = 1e-08$. And CUDA is used to speed training up.

4.4 Evaluation Measure

The official score is based on the F1 for the positive class for subtask 1. As for subtask 2, the official score is based on the macro-averaged F1 of the evaluated classes. The evaluated classes include: Term, Alias-Term, Referential-Term, Definition, Referential-Definition, and Qualifier. Figure 2 (b) shows the proportion of these evaluated classes, we can find the 'I-definition' which occupy a high proportion. The distribution of develop set tags are consistent to train set.

5 Result and Analysis

In this section, we show the performance about subtask 1 and subtask 2 respectively and do some ablation experiments for subtask 1.

5.1 BERTatDE at Subtask 1

we do some experiments as Table 1 shows. RCNN consists of bidirectional LSTM and a layer of maximum pooling. DPCNN proposed by (Johnson and Zhang, 2017), and CNN is composed by three convolutional layers. By comparing the F1 value, we find our model is effective. If our model removes the attention layer, the F1 score from 0.797 to 0.772. And if our model replace the part other than BERT, we can see the F1 is reduced to a certain extent. This result is in the rank of 18 in the official ranking list.

result	model	F1	precision	Recall
Subtask1	BERT+BiLSTM	0.7720	0.7924	0.7527
	BERT+BiLSTM+Attention	0.7971	0.7985	0.7960
	BERT	0.7065	0.6931	0.7204
	BERT+RCNN	0.7853	0.8045	0.7670
	BERT+CNN	0.7933	0.8256	0.7634
	BERT+DPCNN	0.7560	0.7868	0.7276
Subtask2	BERT+BiLSTM+CRF	0.4073	0.4057	0.5028

Table 1: Experimental results about subtask 1 and subtask2

5.2 BERTatDE at Subtask 2

By analysing the data of subtask 2, we find the proportion of official evaluation classes are small as Figure 3 (b) shows. There is an imbalance between different tags. Although the precision is above 85%, the macro-averaged F1 score in official list is not very high. When we training our model, we use the learning rate which is 2e-5. The result of subtask 2 as Table 1 shows.

[3] https://pypi.org/project/pytorch-transformers/
[4] https://pypi.org/project/pytorch-crf/0.7.2/

6 Conclusion

In this paper, we introduce BERTatDE. For multi-domain problem, we make use of BERT to solve this problem. And we use BERT, BiLSTM and attention to classify sentences, we use BERT, BiLSTM and CRF to sequence labeling. By analysing the data of subtask 2, we find the impact of imbalance between the tags. We will use multi-task learning to solve subtask 1 and subtask 2 according to the requirement of definition extraction and we will explore how to alleviate the impact of the imbalance in the future. Considering that there are connections between some tags, we'll learn more about how to use the correlation between these tags to improve our model.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61572120) and the Fundamental Research Funds for the Central Universities (No.N181602013).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luis Espinosa-Anke and Horacio Saggion. 2014. Applying dependency relations to definition extraction. In Elisabeth Métais, Mathieu Roche, and Maguelonne Teisseire, editors, *Natural Language Processing and Information Systems*, pages 63–74, Cham. Springer International Publishing.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. pages 562–570, 01.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition.

Adam Przepiórkowski. 2008. *Definition Extraction with Balanced Random Forests*. Springer Berlin Heidelberg.

Francesco Ronzano. 2015. Weakly supervised definition extraction. In *Recent Advances in Natural Language Processing (RANLP 2015)*.

Steven Schockaert. Syntactically aware neural architectures for definition extraction.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics.

Sasha Spala, Nicholas A. Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. DEFT: A corpus for definition extraction in free- and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131, Florence, Italy, August. Association for Computational Linguistics.

Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. Semeval-2020 task 6: Definition extraction from free text with the deft corpus. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2019. A joint model for definition extraction with syntactic connection and semantic consistency. *CoRR*, abs/1911.01678.

E. N. Westerhout. Definition extraction using linguistic and structural features.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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)*, pages 207–212, Berlin, Germany, August. Association for Computational Linguistics.

Špela Vintar. 2012. Nlp workflow for on-line definition extraction from english and slovene text corpora. *Information Retrieval & Textual Information Access*, pages 53–60.