A Two-stage Algorithm For Chinese Short Text Entity Recognition and Linking

Jieting Li[1] and Tao Jiang[2]

[1] College of Computer Science and Technology
Zhejiang University, Zhejiang 310007, China
lijieting@zju.edu.cn
[2] School of Compute and Information Engineering
Hefei University of Technology, Anhui 340100, China
jiangtao_hf@126.com

Abstract. Entity Recognition and Linking (ERL) for Chinese short texts is one of the basic tasks in the Natural Language Processing (NLP), which aims to detect the mentions in a given Chinese short text and link them with the corresponding entities in a given knowledge base. The entire process of ERL includes two subtasks: mention detection (MD) and entity disambiguation (ED). Due to the diversity of mention and the lack of contextual information in Chinese short texts, it brings new challenges to the ERL task. In order to overcome these difficulties, we propose a two-stage algorithm. Fisrt, we integrate traditional word embedding based models and novel BERT based models to effectively identify possible mentions for MD task. For Chinese short text ED task, we consider it as a ranking problem and propose one pointwise ranking method, which incorporates semantic similarity with entity popularity. In the evaluation of the CCKS 2019 shared task ERL, our model achieves 0.7859 in the F1 score.

Keywords: Entity Linking, Entity Recognition, Word Embedding, BERT.

1 Introduction

Entity Recognition and Linking (ERL), which maps entities in documents to a given knowledge base (KB), plays a very interesting foundation in many areas, such as question answering, semantic search, and information extraction. In recent years, English ERL technology is rapidly developing, and a number of relatively mature English ERL systems have emerged, such as AIDA [1] (Accurate Online Disambiguation of Named Entities) developed by Max Planck Lab in Germany based on YAGO Knowledge Base, DBpedia Spotlight [2] developed by DBpedia.org, etc.

The traditional ERL task is mainly for long documents, and the contextual information owned by long documents can assist entity disambiguation. In contrast, ERL of Chinese short texts has great challenges. The main reasons are as follows: (1) serious colloquialism, which makes entity disambiguation difficult; (2) short text context is not rich in context, a precise understanding of the context is required; (3) compared with English, Chinese is more challenging in the EL problem for short texts due to the characteristics of this language. Therefore, we cannot apply the long text ERL method to this task.

In this paper, we propose a two-stage method for the ERL shared task in CCKS 2019. For MD task, we integrated traditional word embedding based models with novel BERT based models to better detect possible mentions. For ED task, we consider it as a ranking problem and use one pointwise ranking method in the Learning to Rank (LTR) algorithm to rank the candidate entities and select the optimal entity. While considering semantic similarity, we incorporate importance often used in ranking problems to enhance the disambiguation model's performance.

The rest of this paper is structured as follows: Section 2 contains related work. In Section 3, we describe our solution for this task. Experimental results and discussions are presented in Section 4, and finally, we give some concluding remarks in Section 5.

2 Related Work

ERL is an important step in the population of the knowledge base. With the rise of knowledge graph, ERL technology has received more and more attention. But previous researchers are more focused on long text ERL systems. As search technology improves, ERL can enhance the search experience. Search texts are often short texts that bring about new challenges to ERL due to the lack of contextual information. Cornolti et al. [3] put the query statement into the search engine to get some short text related to it, and then extract the related entities from these short texts. This idea is to solve the problem of query noise and lacking contextual information by means of search engines. Deepak et al. [4] proposes to put short text into Wikipedia for a query, and obtain the most relevant k sentences in Wikipedia according to the default ranker provided by Lucene (an open source full-text search engine toolkit). Then entities are extracted from these sentences as candidate entities and 18 features are designed to train a regression model to rank the candidate entities. Nie et al. [5] proposed a new neural network framework. Based on the representation and interaction-based neuro-semantic matching model, the semantic information between the local context and the candidate entity is obtained, and then the ranking aggregation mechanism combines two matching signals for disambiguation.

In addition to ED, MD is also an important step in ERL that may affect the performance of disambiguation. The combined model of bidirectional long short-term memory Bi-LSTM and conditional random field CRF is one of the most classic models in NER, which can also be used for MD. BERT, or Bidirectional Encoder Representations from Transformers [6], a new method of pre-training language representations which obtains state-of-the-art results on a wide array of NLP tasks, with becoming one of the popular models in NLP. Later, there have been many variants of BERT, including ERNIE [7] proposed by Baidu and BERT-wwm [8] proposed by Harbin Institute of Technology. The purpose of ERNIE and BERT-wwm is to solve the shortcomings of pretraining in Chinese corpus and improve the performance of downstream tasks. According to the original paper of BERT, they adopt a feature-based approach to solve NER problem, that is, extract the activations from one or more layers without fine-tuning any parameters of BERT. And the NER model of BERT and Bi-LSTM get the comparable performance than the state-of-the-art algorithms.

3 Model Description

3.1 Mention Detection for Short Text

Mention detection (MD) is the first step in our ERL system. The target of this stage is similar to the named entity recognition (NER), which belongs to the sequence tagging problem. The difference is that NER system tags person, place names and etc., and the mention detection tags the entity (the concrete and unique object that exists objectively, cannot be divided, such as "中国科学技术大学") and concepts (including the category concept "电影", the predicate or attribute concept "妻子", the event concept "美国大选" and other abstract concepts "上午").

Due to the diversity of the tagging content, we try to build several commonly used sequence tagging models, including word embedding based models (the Bi-LSTM model and DGCNN model), and the BERT based models (the novel BERT/ERNIE/BERT-wwm+Bi-LSTM+CRF model), and finally combine the results of these models to improve precision as possible without losing recall.

The form of tagging Mention detection can be regarded as a problem of sequence tagging like NER, and there are many forms of tagging which used in NER commonly, such as pointer form [10], BIO form, BMEWO form. These are all character tagging methods, that is, the input is based on characters, and it is possible to ensure that boundary segmentation errors are avoided as much as possible. In word embedding based models we will use the pointer tagging form, and in BERT based models, we will use BIO and BMEWO tagging form.

3.2 Word Embedding Based Models for MD

We adopted two word embedding based models, namely Bi-LSTM and DGCNN (Dilate Gated Convolutional Neural Network) [10].The input to the model is word embedding, and the output is the probability of a word as the head of mention and the probability of a word as the end of the mention. In other words, we use pointer tagging form. The word embedding we used came from Tencent AI Lab[1]. The reason why we did not use CRF layer here is that the pointer tagging form does not need to ensure the consistency of the tagging content like traditional NER.

Combine character-based and word-based representations (CWR) In simple models, in order to avoid the boundary segmentation error, we should choose character tagging. However, the simple character-based embedding is difficult to store effec-

[1] https://ai.tencent.com/ailab/nlp/embedding.html

tive semantic information. In other words, a single character is basically without semantics, and a scheme for more effectively incorporating semantic information should combine character-based and word-based representations. Specific implementation can refer to [9].

Feature engineering We use feature engineering in word embedding based models, such as Bi-LSTM and DGCNN, in an attempt to improve the performance of the annotations. The features are listed below: (1) Word Segmentation feature f_1: f_1 is proven to be effective in [11]. We use jieba[2] tool to segment the text, and perform simple labeling rule on the text after the word segmentation. (2) Lexicon feature f_2: most state-of-the-art NER systems make use of lexicons as a form of external knowledge. We use the lexicon coding scheme in [12] to construct lexicon feature, that is, we match every n-gram (up to the length of the longest lexicon entry) against entries in the lexicon. (3) Position feature f_3: we equip our models with a sense of order by embedding the absolute position of input elements refer to [13]

Modify training set When we examined the dataset, we found that there were some gaps and irregularities in the training set. In order to improve the quality of the training set, we use nine-fold cross-validation method to obtain nine models, and then infer the training set back to get nine prediction results. If one mention appears in nine predictions at the same time but does not appear in the original training set, then the mention is added to the tagging result of the sample; if it doesn't appear once in the nine predictions but are marked by the training set, then the mention is removed from the tagging results of the sample. Using this modified training set to retrain word embedding based models, we will get two different models for model integration.

Modify lexicon When constructing lexicon features, the lexicon is generated by 'subject_alias' and 'subject' in KB, and then mention detection model is trained. When during the inference, a large number of words in the lexicon will decentralize model's attention, so we will summarize the mentions that have appeared in the training set to form a new lexicon, which helps to construct the dictionary features when predicting. We found that after modifying the lexicon, the model was able to find some mentions that appeared less frequently, such as program name, book name, etc. So, we union the two results here with the original word embedding based models' results respectively.

3.3 BERT Based Model for MD

Since BERT appeared in 2018, pre-trained models have become very popular. BERT based models include ERNIE proposed by Baidu and BERT-wwm proposed by Harbin Institute of Technology. Their differences mainly lie in the different methods of the masking during pretraining. The latter two models are more in line with the lin-

[2] https://github.com/fxsjy/jieba

guistic characteristics of Chinese. There are two methods called fine-tuning and feature extraction to use these pre-trained models for downstream tasks. In order to ensure the efficiency of these models, we only use feature extraction with task-specific model architecture, that is, we input the output of the last layer of these pre-trained models into our next Bi-LSTM+CRF model. We adopt BIO and BMEWO forms for tagging respectively. Experiments have shown that the BMEWO form tends to find longer mention.

3.4 Ensemble Learning for Mention Detection

In order to deal with the recognition problems caused by the diversity of mention, we integrate these above models. Bagging, also known as bootstrap aggregating, which is to obtain T new data sets after selecting T times in the original data set, and we use the simple voting method in bagging for result integration. When the number of votes about one tagged mention is greater than a certain threshold t, we accept it, otherwise we reject it. On the validation set, we tested the voting performance and found that the performance of $t = 5$ of ten predicted results is the best.

3.5 Entity Disambiguation

Entity disambiguation (ED), designed to link mentions tagged in the MD stage to the KB, can be seen as a ranking problem. So we can use the pointwise algorithm in LTR like logistic regression (LR) to solve this problem. We first encode the semantics that need to be matched, then integrate features into LR.

The general approach to entity disambiguation includes two steps. The first step is to find the candidate entities and encode mention semantics and candidate entity semantics. For efficiency considerations, we only use Bi-LSTM to encode the semantics. The contextual information of mention is encoded together with mention as mention semantics, and all the description texts of the entity are spliced together and encoded together with entity as entity semantics. The second step is to calculate the similarity between mention and candidate entity according to the semantic encoding, either by using the distance formula directly or through a simple forward neural network (FNN). We used the second method to dynamically train FNN. Below are some of the details of the model.

Candidate Entities Generation Because the mention m must appear in the subject or alias field of the KB, we only need to count all subject and alias information $\{e_n\colon [m_{n,1}, m_{n,2}, m_{n,3}, \ldots]\}$ to construct a map $\{m_n\colon [e_{n,1}, e_{n,2}, e_{n,3}, \ldots]\}$. After that, we can quickly find candidate entities based on the mention.

Semantic Encoding We use Bi-LSTM to capture the contextual information of the input character sequence, and then go through a maxpool layer along the sequence direction to get a fixed-dimensional vector to represent the semantic encoding of a sentence.

Feature engineering. (1) Contextual feature f_4: After segmenting sentence and removing stop words, the normalized proportion of the same entity between query text and entity description is calculated as the contextual feature; (2) Type feature f_5: The type information existing in the KB is relatively regular and there are 51 categories about it, and we embed them as type feature. (3) Entity popularity f_6: We regard ED as a ranking problem, and then rank the candidate entities to select the best entity to be linked. Ranking algorithms generally consider both similarity and importance. In this model, we simulate similarity through semantic encoding and FNN, while importance is not considered. Our solution is to crawl the number of related results about each entity in the Baidu search, and take the logarithm of the number as the importance of the entity, or called entity popularity.

4 Experiments

4.1 Datasets and Implementation

The data set provided by CCKS 2019 contains 90,000 short text training set data and 30,000 short text test set data. The entity data of KB comes from Baidu Encyclopedia, with 399,252 pieces of entity information. We finally use the F1 score as the evaluation index for MD stage. For a given Chinese short text query, the output of the MD system contains all mentions appearing in the given Chinese short text query. We calculate the precision, recall, and F1 score by comparing the output to the gold set. For ED stage, we use accuracy as an evaluation indicator.

4.2 Mention Detection Results

Table 1 gives the results of the word embedding based models with the feature engineering. As can be seen from Table 1, the features f_1 and f_2 play an important role in the MD stage. After adding CWR and f_3, our model has achieved excellent performance and F1 score can reach 0.8146. Table 2 is the MD result of BERT based models. Due to BERT based models pretrained on a large number of non-labeled corpora and the advanced structure of these models, even if the feature engineering is not performed, good results can be obtained. In addition, we found in the experiment that the model of the BMEWO tagging form can dig more long mentions than the model of BIO tagging form.

Table 1. Experimental results of mention detection about word embedding based models and feature engineering

Model	F1	Precision	Recall
Bi-LSTM	0.7099	0.7092	0.7106
Bi-LSTM+$\boldsymbol{f_1}$	0.7859	0.7811	0.7907
Bi-LSTM+$\boldsymbol{f_1}$+$\boldsymbol{f_2}$	0.8030	0.8100	0.7961
Bi-LSTM+$\boldsymbol{f_1}$+$\boldsymbol{f_2}$+CWR	0.8101	0.8162	0.8041
Bi-LSTM+$\boldsymbol{f_1}$+$\boldsymbol{f_2}$+CWR+$\boldsymbol{f_3}$	0.8121	0.8234	0.8010

DGCNN+$\boldsymbol{f_1}$+$\boldsymbol{f_2}$+CWR+$\boldsymbol{f_3}$	0.8146	0.8103	0.8189

Table 2. Experimental results of mention detection about BERT based models

Model	F1	Precision	Recall
BERT+BIO	0.7942	0.7847	0.8039
ERNIE+BIO	0.7931	0.8013	0.7850
BERT-wwm+BIO	0.8002	0.8028	0.7977
BERT+BMEWO	0.8014	0.7934	0.8100
ERNIE+BMEWO	0.7918	0.7992	0.7847
BERT-wwm+BMEWO	0.8025	0.8029	0.8021

4.3 Entity Disambiguation Results

Table 3 shows the experimental results about ED model LR with feature engineering.

Table 3. Experimental results of LR and feature engineering

Model	Accuracy
LR	0.8886
LR+$\boldsymbol{f_4}$	0.8997
LR+$\boldsymbol{f_5}$	0.8982
LR+$\boldsymbol{f_4}$+$\boldsymbol{f_5}$	0.9059
LR+$\boldsymbol{f_4}$+$\boldsymbol{f_5}$+$\boldsymbol{f_6}$	0.9073

Because we have eliminated some mentions of missing candidate entities in MD phase, the final result is a little better than table 3. According to competition officials, 2.592% of the entities could not find a candidate because their names did not match with the subject or alias information in the KB.

We submitted the final result in the evaluation of the CCKS 2019 shared task ERL, which reached the F1 value of 0.7859.

5 Conclusion

In this paper, we consider Chinese short text ERL task of CCKS 2019 as two subtasks, mention detection and entity disambiguation and propose a two-stage solution. For mention detection, we integrated traditional word embedding based models with novel BERT based models and achieved good performance. For entity disambiguation, we construct the entity popularity features through the relevant results' number returned by Baidu search, and then combine other features as input to one pointwise LTR model for entity disambiguation. In the future, we will pay more attention to the efficiency of the disambiguation model to meet the actual needs.

References

1. Yosef, M. A., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables. In: Proceedings of the VLDB Endowment, vol. 4, pp. 1450-1453 (2016).
2. Mendes, P. N., Jakob, M., García-Silva, A., Bizer, C.: A. DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th international conference on semantic systems, pp. 1-8. ACM (2011).
3. Cornolti, M., Ferragina, P., Ciaramita, M., Rüd, S., Schütze, H.: A piggyback system for joint entity mention detection and linking in web queries. In: Proceedings of the 25th International Conference on World Wide Web, pp. 567-578. International World Wide Web Conferences Steering Committee (2016).
4. Deepak, P., Ranu, S., Banerjee, P., Mehta, S.: Entity linking for web search queries. In: European Conference on Information Retrieval, pp. 394-399. Springer, Cham (2015).
5. Nie, F., Zhou, S., Liu, J., Wang, J., Lin, C. Y., & Pan, R. Aggregated semantic matching for short text entity linking. In: Proceedings of the 22nd Conference on Computational Natural Language Learning, pp. 476-485. (2018).
6. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171-4186. (2019).
7. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Wu, H.: ERNIE: Enhanced Representation through Knowledge Integration. arXiv preprint arXiv:1904.09223 (2019).
8. Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., Hu, G.: Pre-Training with Whole Word Masking for Chinese BERT. arXiv preprint arXiv:1906.08101 (2019).
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition. In: Proceedings of NAACL-HLT, pp. 260-270. (2016).
10. Su, J. L.: A Hierarchical Relation Extraction Model with Pointer-Tagging Hybrid Structure. Github, https://github.com/bojone/kg-2019 (2019)
11. Luo, W. C., Yang, F., An Empirical Study of Automatic Chinese Word Segmentation for Spoken Language Understanding and Named Entity Recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 238-248. (2016)
12. Chiu, J. P., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 4, 357-370 (2016).
13. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y. N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1243-1252. JMLR. Org (2017).