Incorporating Part-of-Speech Feature and Entity Embedding for Question Entity Discovery and Linking

Shijia E, Li Yang, Shiyao Xu, Shengbin Jia, and Yang Xiang

Tongji University, Shanghai 201804, P.R. China,
{436_eshijia,1452238,1452221,shengbinjia,shxiangyang}@tongji.edu.cn

Abstract. Question entity discovery and linking (QEDL), which aims to extract the named entities associated with the given question. Typically, the name of an entity has a certain ambiguity, i.e. the same entity name may refer to multiple entities. In this paper, we propose a model which based on the part-of-speech (POS) feature and entity embedding to solve the QEDL task. The proposed model does not depend on much feature engineering. Our experiments show that the entity embedding can make full use of semantic information involved in an entity and its context word. In the evaluation of the CCKS 2017 shared task, our model achieves 0.5960 in the F1 score of mentions, and 0.3694 in the F1 score of entities for the opening test data.

Keywords: entity linking, entity discovery, entity embedding

1 Introduction

Named entity linking (NEL) is to link a given mention to an entity in the knowledge base. It has been pay attention with the development of natural language processing (NLP) and knowledge graph (KG). In traditional NEL tasks, the text corpus is long text. Therefore, the solutions can use a lot of context features with the mentions to accomplish the entity linking. However, in this shared task, question entity discovery and linking (QEDL) introduced by CCKS 2017, we need to find the mentions and link them to the entities in an existing knowledge base with the short question text. Compared with the traditional NEL tasks, the resources that we can use in QEDL task are just the word in the question and the entity attributes in the knowledge base. Due to the short length of the text in this task, we do not have much context feature of the mentions in the question. Thus, we cannot apply tradition NEL methods directly to this task.

In this paper, we propose a method for the QEDL shared task in CCKS 2017. For the discovery of mention, we use the part of speech features and a variety of combinatorial strategies to effectively identify possible mentions. As for the entity linking, we utilize the entity embedding that is generated based on the entity attributes to do the entity disambiguation. Also, the results of entity linking can help us do the post-processing to improve the performance of mention discovery.

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

2 Related Work

In recent years, many researchers have begun to focus on NEL in short texts, in particular for the Chinese language. Shen et al. [6] propose a method to model Twitter users' data. It can make the candidate entities with similar user interest have a high weight. Guo et al. [1] model the micro-blogs with similar themes to disambiguate candidate entities. To utilize context features, Liu et al. [3] use the similarity between the mention context and entity to accomplish the entity linking with micro-blog data. Jiang et al. [2] use Twitter's forward, reply, and other messages of the same user to expand the context of sentiment classification. The core of the methods mentioned above is trying to use the contextual information needed by the entity linking so that the overall performance can be improved. However, it adds to the cost of data pre-processing and not all data sources have enough contextual information.

As a result, for this QEDL task, we have tried several methods just based on the knowledge base to generate the entity embedding. It contains the semantic information embodied in entity attributes and can be used to do the entity disambiguation with the limited contextual information of mentions.

3 Model Description

In this section, we describe the proposed method to solve the QEDL task. Figure 1 shows the overall framework of our method. We will provide details for each module in the following sections.

3.1 Word segmentation for the question text

Word segmentation is the first step of our system. We use Jieba[1], the Chinese word segmentation tool, to help us get words from each question. Other segmentation tools such as Ansj[2] and Thulac[3] are attempted, but both of them perform weaker than Jieba. We adopt *accurate mode* instead of *all mode* to segment the questions. Thus each word is a substring of a question and not overlapped by others. However, Jieba cannot recognize some relatively long and complicated words accurately. For example, we expect the word "注册会计师" (certified public accountant) could be cut correctly from the question "注册会计师的审计责任包括哪些?" (what are the audit responsibilities of certified public accountants?),

[1] `https://github.com/fxsjy/jieba`
[2] `https://github.com/NLPchina/ansj_seg`
[3] `http://thulac.thunlp.org`

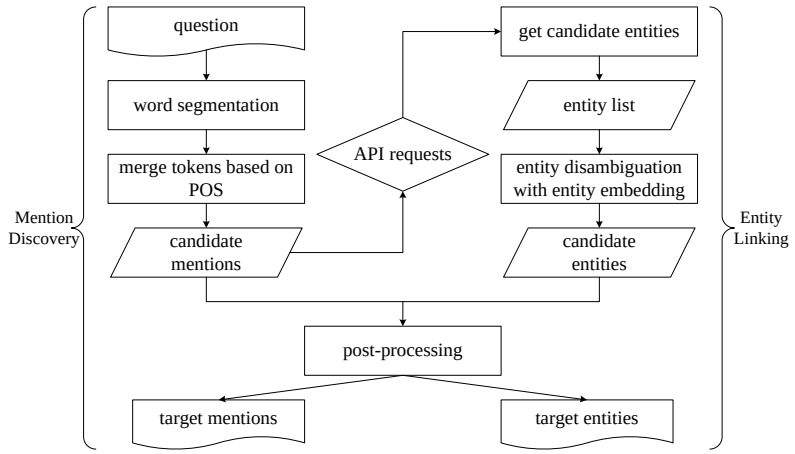

Fig. 1. The overall framework of our proposed method.

but unfortunately, it will be further divided into two words ("注册" (register), "会计师" (accountant)). This problem can be solved by specifying our custom dictionary to be included in the Jieba default dictionary. The custom dictionary contains named entities, such as "小米手机" (Mi phones), "韵达快递" (Yunda Express), from various domains. Each word cut from a question is considered as a candidate mention or a possible part of a candidate mention. We tag the part-of-speech (POS) of each word, which will be used to judge whether the word is a part of a candidate mention.

3.2 Candidate Mention Generation

Merge adjacent words After word segmentation, there are two lists, $Seg = \{a_1, a_2, \ldots, a_n\}$ and $Pos = \{p_1, p_2, \ldots, p_n\}$ (n=the number of words). In most case, Jieba splits one mention into several words (such as "百度知道企业平台" (Baidu Zhidao enterprise platform) becomes "百度知道, 企业, 平台" (Baidu Zhidao, Enterprise, Platform)). We merge all 2, 3, . . . , n-1 and n adjacent words in Seg into one string. These words and new strings are added to initial mention list so that the correct mentions are sure to be preserved. The initial candidate mention set is:

$$Mention_{init} = Seg \cup \{m_{ij} \mid m_{ij} = a_i \ldots a_j (i = 1, \ldots, n-1; j = i+1, \ldots, n)\} \quad (1)$$

Filter based on the POS In our experiments, we can find out that the majority of the mentions of questions are nouns. The parts of speech that represent nouns can be picked out with Jieba. The noun POS set is:

$$Noun\_POS = \{an, ng, n, nr, ns, nt, nz, un, nz, eng, nrt, l, i, j, x\} \quad (2)$$

There is also a stop word list to reduce noise. Only the candidate mentions in the initial list, do not appear in the stop word list and contain at least one word whose POS is in $Noun\_POS$ list are retained. The filtered candidate mention set is:

$$Mention = Mention_{init} - \{m_{ij} \mid m_{ij} = a_i \dots a_j (\forall p_k \notin Noun\_POS, k = i, \dots, j)\} - \{m_{ij} \mid m_{ij} \in Stopwords\} \quad (3)$$

3.3 Get Candidate Entities and Prepare the Entity Corpus

In this task, we must use the CN-DBpedia [7] as the standard knowledge base system. With the API provided by the system, we can get the entity list corresponding to a mention. The target entity will be selected from that list.

Also, the API provided by the knowledge base system allows us to obtain the relevant attributes of entities, such as the description and category of an entity. We can use an entity name and its related attributes to form a line of text. Therefore, we can get a large entity corpus with the candidate entities. This corpus can be used to train the word embedding with word2vec [4]. In this task, we use Gensim [5] to train the word embedding. The embedding size is 300.

3.4 Entity Embedding

For the word segmentation of the entity corpus, we do not cut the entity name so that the entity name will be reserved as a single word. To generate the entity embedding, we first load the word embedding produced by Gensim. For each line in the entity corpus, the first word is the entity name, and the other words are entity attribute values. As a result, an entity embedding is just the average vector of the distributed representations of its attribute values, and the embedding size of an entity is still 300. It is important to emphasize that the embedding produced by Gensim will then be used to generate vector representations of the words in a question.

3.5 Entity Disambiguation

When we get the entity embedding, we can do the entity disambiguation with the candidate entities. The word segmentation results of the question text can be denoted as $q_{words} = \{q_1, q_2, \dots, q_x\}$ (x=the number of words) which are the only contextual information we can use. Therefore, we use the following entity score to represent the similarity between a candidate entity (denoted as ent)and the question:

$$Entity\_Score = \frac{\sum_{k=1}^{x} Cos(Emb(q_k), Emb(ent))}{x} \quad (4)$$

where $Cos(a, b)$ means the cosine similarity of a and b, and $Emb(\cdot)$ means to get the vector representation of the word (from raw word embedding) or entity (from entity embedding). For all the candidate entities of a mention, we select the one that has the maximum $Entity\_Score$ as the entity linked to the mention.

3.6 Post-processing

After the above processing steps, we have got the mentions and entities that meet the requirements. To further improve the performance of the proposed method, we need to do the necessary post-processing. First of all, in the candidate mention list, if $Mention\_A$ is included in $Mention\_C$, e.g. "百度知道" (Baidu Zhidao) ∈ "百度知道企业平台" (Baidu Zhidao enterprise platform), then $Mention\_A$ and its corresponding entity will be removed from the final results. In addition, if $Mention\_B$ does not have a linked entity in the knowledge base, and $Mention\_B$ is not in the dictionary, then it will be removed as well.

4 Experiments

4.1 Datasets and Implementation

We apply the proposed method directly to the QEDL task. The train set provided by CCKS 2017 shared task contains 1,400 questions, and the test set contains 749 questions. We use the train set to improve the stability of the model and extend the dictionary. For the embedding size, we tried 50, 100, 300 and 500, and the size of 300 performed best on the train set. Therefore, we use the best parameter configuration on the train set for the final evaluation.

For the entity disambiguation, we also tried to use a similarity measure based on the longest common subsequence (LCS). Specifically, we calculate the length of the LCS between the question and the entity description retrieved by the API and then select the entity with the maximum length of LCS as the target entity.

4.2 Results

Table 1 shows the experimental results on the train set and test set in this task.

As the results shown in Table 1, the proposed method with the entity embedding achieves a good performance with the F1-Entity of 0.3694 on the test set. It yields a significant improvement compared to the method with LCS. It proves that the generated entity embedding contains valid semantic information, which can be used to compare the similarity with the context of mentions, so as to complete the entity disambiguation task effectively.

Table 1. The experimental results for the train set (top) and test set (bottom).

Method	F1-Mention	Precision-Mention	Recall-Mention	F1-Entity	Precision-Entity	Recall-Entity
POS + LCS	0.5153	0.4347	0.6327	0.3353	0.2828	0.4117
POS + Entity embedding	0.6839	0.6120	0.7750	0.4942	0.4422	0.5600
POS + LCS	0.5114	0.4295	0.6320	0.2805	0.2284	0.3635
POS + Entity embedding	0.5960	0.5103	0.6681	0.3694	0.3189	0.4388

5 Conclusion

In this paper, we describe the method which incorporates the POS feature and entity embedding for the QEDL task. Our method can achieve good performance with few feature engineering. For future work, we need to study more effective evaluation metrics to test whether the results of the mention discovery and entity linking can meet the requirement of practical applications.

Acknowledgments This work was supported by the National Basic Research Program of China (2014CB340404), the National Natural Science Foundation of China (71571136), and the Project of Science and Technology Commission of Shanghai Municipality (16JC1403000, 14511108002).

References

1. Guo, Y., Qin, B., Liu, T., Li, S.: Microblog entity linking by leveraging extra posts. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 863–868 (2013)
2. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: ACL (1). pp. 151–160. Association for Computational Linguistics (2011)
3. Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., Lu, Y.: Entity linking for tweets. In: ACL (1). pp. 1304–1311 (2013)
4. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
5. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer (2010)
6. Shen, W., Wang, J., Luo, P., Wang, M.: Linking named entities in tweets with knowledge base via user interest modeling. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 68–76. ACM (2013)
7. Xu, B., Xu, Y., Liang, J., Xie, C., Liang, B., Cui, W., Xiao, Y.: Cn-dbpedia: A never-ending chinese knowledge extraction system. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. pp. 428–438. Springer (2017)