Open Domain Named Entity Discovery and Linking Task

Yeqiang Xu, Zhongmin Shi(✉), Peipeng Luo, and Yunbiao Wu

[1]Summba Inc., Guangzhou, China
{yeqiang, shi, peipeng, yunbiao}@summba.com

Abstract. This paper describes a named entity discovery and linking system, which compete the CCKS2017 question named entity discovery and linking task. We are facing challenges including short-text, small training samples and open domain, making the existing solutions unfeasible. In this paper, we propose a CRF + rules method to recognize the corresponding named entity, which employs several features, such as bag-of-word features, POS features, parsing features etc. As for entity linking, context information, popularity, word embeddings, and online public corpus are used. The experiment results show that, the F1 score of named entity discovery is 0.815, while the accuracy of the entity linking is 0.736. The overall F1 score is 0.600, which proves the effectiveness of our system.

Keywords: Named Entity Discovery, Entity Linking, Context Information.

1 Introduction

With the development of Internet techniques, unstructured text has become one of the most popular information carriers. As the basis of text analysis, Named Entity Discovery (NED) and Entity Linking (EL) are widely studied. However, the traditional NED techniques can only be applied to the situation with very few entity types (e.g. persons, locations, organizations etc.), and the existing EL methods need rich context information. In this task, we mainly encounter three difficulties. Firstly, the boundary of Named Entity (NE) definition is fuzzy. For example, “苹果手机” (Apple iPhone) is not a named entity, but “苹果” (Apple) is a named entity as a company name; The second challenge is that the questions in training set are short-texts with lots of noises; The last challenge is that the corpus is from open domain and entity types are relatively much more.

2 Related Work

Our work can be divided into two parts: NED and EL. The existing approaches of NED are mainly based on statistical models, e.g. HMM (Hidden Markov Model), CRF (Conditional Random Field) and DNN (Deep Neural Network) etc[1][2][3]. Most systems of EL adopt supervised methods to disambiguate, including binary

classification[4][5] and Machine-Learned Ranking-based EL (MLR-based EL)[6][7]. Some studies suppose NED and EL are related sub-task, and achieved a high performance by optimized the joint model[5].

However, the statistical models of NED principally focus on very few entity types, it is unsuitable in open domain. As for EL, supervised approach is difficult to apply, with short-text and limited information about each sense of ambiguous named entity.

3 Named Entity Discovery & Entity Linking

3.1 Named Entity Discovery

Our strategy is CRF + rules in NED. Firstly, bag-of-word features and POS features from the training corpus are extracted with the pyltp tool[8]. The training corpus is then transformed into character-level corpus, and divided into training set and test set. Also, it is necessary to further filter, split and re-identify them based on rules.

Rule 1: Filter Rules
The filter rules are divided into the following categories (more than 180 cases are included, while the size of training set is 1395):

a) Version number filtering: the product number should be included as part of the named entities, while the version number should not.
b) Suffix filtering: some suffixes affecting NED must be filtered. For instance, for the phrase "戴尔笔记本" (Dell Laptop), the real NE is "戴尔" (Dell), while "笔记本" (Laptop) is a generic entity cannot be included.
c) Verb-prefix filtering: in the prediction results, there are very few named entities which has structure of "verb + noun", in which case the verb prefix needs to be filtered out.

Rule 2: Split Rules
Among the predicted named entities, if conjunctions exist, such as "and" etc., the results need to be split into more parts.

Rule 3: Re-identify Rules
The CRF model-based result can be revised by re-identify rules. In the beginning, a named entity dictionary in the training set is constructed. By using the dictionary, we re-identify a new result based on all-matching rule. If the re-identified result does not overlap with the model-based result, it can be added to the final result set. In the case of overlap, when the model-based result is contained in re-identified result, the re-identified result should be added to the final result set. In other cases, the result is based merely on the model-based result. Just as one example, assuming the model-based result is “百度知道” (Baidu Zhidao), while the re-identified result is “百度知道企业平台” (Baidu Zhidao Enterprise Platform), the final result can be revised as “百度知道企业平台” (Baidu Zhidao Enterprise Platform).

3.2 Entity Linking

Several useful features can be used for EL, such as entity context information in original sentence, popularity of each entity's sense, word co-occurrence in Baidu Zhidao[1] for each entity's sense etc.

Information Extension
In this task, information extension is necessary for the reason that each ambiguous named entity has several senses with less information. We collect each sense's relative questions as candidate question sets by searching Baidu Zhidao and picking out the questions in top 5 pages. For example, a sense from knowledgeworks[2] calling "小米（小米公司）" (Xiaomi Inc.) is extended to "小米公司的经营理念是什么?" (What is the business philosophy of Xiaomi Inc.) as one candidate question by searching Badidu Zhidao.

Information Filter
Firstly, all questions are segmented. Then stop words are removed. At last, the similarity of original question with each candidate question is calculated by using Jaccard Similarity, and those candidate questions with low similarity are removed.

Context Similarity
The original and candidate questions need to be represented as vectors. In a question, each word's vector representation is gained by pre-trained word2vec[9] model. After filtration, a sentence has only a few words left, therefore, we simply average word vectors in a sentence. Let $W = \{w_i\}$ be a word vector set whose size is N. Then the question's sentence vector v is presented as formula (1).

$$v = \frac{1}{N} \cdot \sum_1^N w_i \tag{1}$$

The context similarity score of an entity's sense can be calculated by the average cosine similarity of the original question with each one in the candidate set of this sense. Let o be the original question's sentence vector, m be one of entity's sense, $V = \{v_i\}$ be the sentence vectors in candidate set of the sense whose size is K. The context similarity score of o and m is shown as formula (2).

$$\text{score}_{\text{context}}(o, m) = \frac{1}{K} \cdot \sum_{l=1}^{K} \text{similarity}_{\text{cosine}}(v_o, v_l) \tag{2}$$

Popularity
We collect every sense's visits of a named entity from Baidu Baike[3] as one index of popularity. But, some of entity's senses in knowledgeworks are different from those in Baidu Baike, in which case edit distance is needed. For those entity's senses never show up in Baidu Baike, we use the minimum score from other entity's senses to ensure score's smoothness. Let o be the original question, M be the entity's senses

[1] https://zhidao.baidu.com/
[2] http://knowledgeworks.cn:30001/
[3] https://baike.baidu.com/

in knowledgeworks, $m \in M$ be one of entity's sense, $k \in M, k \neq m$ be another entity's sense different from m, S be the size of senses in Baidu Baike, "editDist" be the edit distance of m and the processing sense in Baidu Baike. The score for an entity's senseis shown as formula (3).

$$\text{score}_{\text{visit}}(o, m) = \max\left(\left(\frac{1}{S} \cdot \sum_{i=1}^{S} \text{visits}_i \cdot \text{editDist}\right), \left(\min_{k \in M, k \neq m} \text{score}_{\text{visit}}(o, k)\right)\right) \quad (3)$$

For each entity, knowledgeworks provides a frequently-used sense calling primary sense. Let t be 1 or 0 presenting whether m is primary, g be the weight when m is not primary for score's smoothness. The primary score is shown as formula (4).

$$\text{score}_{\text{primary}}(o, m) = g^{1-t} \quad (4)$$

Word Co-occurrence

After processing irrelevant information filtration and Chinese segmentation for extended questions, word co-occurrence frequencies are calculated by counting each word shown up together in the two extension sets of original questions and candidate ones in the entity's senses, and then normalized. Let o be the original question, m be one of entity's sense whose size is N, $\text{count}_i, i \in N$ be the word co-occurrence of arbitrarily one of the entity's senses. The score is shown as formula (5).

$$\text{score}_{\text{co-occ}}(o, m) = \frac{\text{count}_i}{\sum_1^N \text{count}}, i \in N \quad (5)$$

EL Scoring Method

According to the four scores described above, the weighting method is shown as formula (6).

$$\text{score}_{\text{final}}(o, m) = \alpha \cdot \text{score}_{\text{context}} + \beta \cdot \text{score}_{\text{visits}} + \gamma \cdot \text{score}_{\text{primary}} + \mu \cdot \text{score}_{\text{co-occ}}$$

$$\text{s.t. } \alpha + \beta + \gamma + \mu = 1 \quad (6)$$

EL Parameters

The following Table 1 *lists EL parameters for formulas described above.*

Table 1. EL parameters.

	editDist	g	α	β	γ	μ
Parameter	2	0.3	0.6	0.1	0.1	0.2

All parameters in Table 1 are adjusted by repeated testing. The editDist means the edit distance of senses. the g presents the weight of non-primary parameter. As for the combination parameters, the sentence context similarity α contributes most to EL; the visit β and primary γ presenting popularity are assigned the same value; the word co-occurrence μ also act as an important role for EL.

4 Evaluation

4.1 Dataset and Pre-training of Word Embedding

In this experiment, we use the dataset provided by CCKS2017 Task 1. Besides we crawl tens of millions of questions from Baidu Zhidao and train a word2vec[9] model.

4.2 Experimental Result

Named Entity Discovery Result
The following Table 2 *lists NED results.*

Table 2.NED results.

Method	Result		
	P	R	F
CRF	0.806	0.703	0.753
CRF+Rule1	0.830	0.708	0.764
CRF+Rule1+Rule2	0.837	0.710	0.768
CRF+Rule1+Rule2+Rule3	0.887	0.754	0.815

Table 2 shows that all rules are effective, and Rule 3 contributes most. Without using rules, the bottleneck of F1 is 0.753, while CRF + rules achieves +0.062 F1 score.

EL Result
The Table 3 *shows EL results.*

Table 3.EL results.

	Context vector	Visits	Primary	Co-occurrence	Combine
P	0.584	0.643	0.548	0.575	0.736

Table 3 shows that the combination of four scores is better than anyone of them. The maximum precision of EL is 0.736. As the Table 1 shows that Context vector affects the result most, which means the importance of context information for EL research.

Final Result
The final experimental result is shown in Table 4.

Table 4. Experimental result.

NED			EL	Over all
P	R	F	P	F
0.887	0.754	0.815	0.736	0.600

According to the result, the best score of NED in training set is based on CRF with

rules, which F1 is 0.815. The top precision of EL comes from the combination of four features, which is 0.736. So as to the overall F1 is 0.600.

5 Conclusion and Future Work

In this paper, we propose some strategies for NED and EL to deal with open domain short-text issues. Experiments show that our method has effective performance. In the future, we are trying to apply NED by using Heuristic and DNN[3] method. As for EL, CNN[5] can be considered as research direction.

Acknowledgments

This work was supported in the Research on People's Heterogeneous Information Aggregation Technology, and Research and Development of Intelligent Question Answering System Based on Text Automatic Abstract Technology, and the Research on Key Techniques of Knowledge Map in Intelligent Home.

References

1. Morwal S, Jahan N, Chopra D. Named entity recognition using hidden Markov model (HMM)[J]. International Journal on Natural Language Computing (IJNLC), 2012, 1(4): 15-23.
2. Zhou, J., Dai, X., Yin, C., Chen, J.: Automatic recognition of Chinese organization name based on cascaded conditional random fields. Acta Electronica Sinica 34(5), 804 (2006).
3. Ma, X., Hovy, E.: End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1064-1074 (2016).
4. Zhang, W., Su, J., Tan, C-L, Wang, W-T.: Entity Linking leveraging automatically generated annotation. In: International Conference on Computational Linguistics (COLING 2010), pp. 1290-1298 (2010).
5. Francis-Landau, M., Durrett, G. and Klein, D.: Capturing semantic similarity for Entity Linking with Convolutional Neural Networks. In: Proceedings of NAACL-HLT, pp. 1256-1261 (2016).
6. Ratinov, L., Roth, D., Downey, D. and Anderson, M.: Local and global algorithms for disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume, pp. 1375-1384 (2011).
7. Shen, W., Wang, J., Luo, P. and Wang, M.: Linden: linking named entities with knowledge base via semantic knowledge. In: Proceedings of the 21st international conference on World Wide Web, pp. 449-458 (2012).
8. Che, W., Li, Z., Liu, T.: LTP: A chinese language technology platform. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 13-16 (2010).
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, pp. 3111-3119 (2013).