A Hybrid Re-ranking Method for Entity Recognition and Linking in Search Queries

Gongbo Tang[1,2], Yuting Guo[2], Dong Yu[1,2(✉)], and Endong Xun[1,2]

[1] Institute of Big Data and Language Education,
Beijing Language and Culture University, Beijing 100083, China
{tanggongbo,yudong_blcu,edxun}@126.com
[2] College of Information Science,
Beijing Language and Culture University, Beijing 100083, China
guoyuting_gyt@126.com

Abstract. In this paper, we construct an entity recognition and linking system using Chinese Wikipedia and knowledge base. We utilize refined filter rules in entity recognition module, and then generate candidate entities by search engine and attributes in Wikipedia article pages. In entity linking module, we propose a hybrid entity re-ranking method combined with three features: textual and semantic match-degree, the similarity between candidate entity and entity mention, entity frequency. Finally, we get the linking results by the entity's final score. In the task of entity recognition and linking in search queries at NLPCC 2015, the *Average-F1* value of this method achieved 61.1% in 3849 test dataset, which ranks second place in fourteen teams.

1 Introduction

Search engine is the most common way to access information, sometimes, people have to search satisfied answer in retrieval result because of information explosion. To return better retrieval result, we need to deal with the entity recognition and linking task to understand users' intents better.

Search queries are nonstandard text, containing wrong spellings and abbreviation, alias names, nick names of entity. For instance, "习大大爱着彭妈妈", obviously, "习大大" refers to "习近平"and "彭妈妈" is nick name of "彭丽媛". Meanwhile, search queries are really short, the longest query may be dozens of words. Compared with the traditional entity recognition and linking works, the context can't provide enough features to disambiguate entities.

In this paper, we show our system for entity recognition and linking in search queries. There are three main stages, entity recognition, candidate entities generation and entities disambiguation. We use the rules and entity base to filter the entity mention, then we generate candidate entities by search engine and attributes of Wikipedia pages. Finally, we utilize an entity re-ranking method to score the candidate entities, which is combined with three features: textual and semantic match-degree, the similarity between candidate entity and entity mention, entity frequency. Figure 1 shows the framework of our system.

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 598–605, 2015.
DOI: 10.1007/978-3-319-25207-0_57

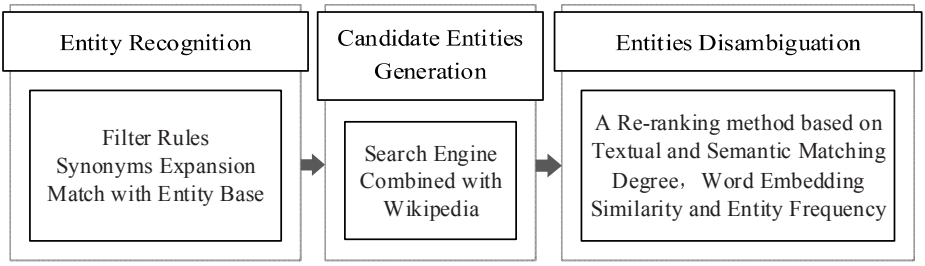

Fig. 1. The framework of our system

2 Related Work

There are two methods to recognize entities, rule based and machine learning based. Methods based on rules always recognize entity by the spelling rules, the parts of speech of entities and dictionary. Mikheev et al. [1] can recognize 79.3% entities by using a general dictionary, as some entities are ambiguous with common nouns, the remaining 20.7% entities are not recognized. In addition, methods based on machine learning always need a lot of tagged dataset to learn a model. For instance, Asahara and Matsumoto [2] utilize a support vector machine model to recognize entities.

To generate candidate entities, Han and Zhao [3] use Google API to get the retrieval result of short text, and select the entities in the title of Wikipedia pages as candidate entities. What is more, Meng et al. [4] utilize Baidu to search entity mention+ "维基百科"and entity mention+ "维基百科", and also select the entities in the title of Wikipedia pages as candidate entities. The candidate entity disambiguation need features, Dalton and Dietz [5] utilize urban dictionary to expand query. Hoffart et al. [6] use the word frequency in Wikipedia to define popularity feature. Blanco et al. [7] consider the distributional semantics of query words and entities, and train word embedding by word2vec. Shen et al. [8] make use of a SVM model, and give a rank to candidate entity for each entity mention with a linear combination of four features: entity popularity, semantic associativity, semantic similarity and global topical coherence between mapping entities.

3 Method

To reduce the system's complexity, we used some refined rules to filter and classify entity mentions. It's simple, but effective! To deal with various names, we use the synonym dictionary to expand entity mentions. And we use the search engine combined with Wikipedia to generate candidate entities. Finally, we utilize a re-ranking method to get the results. Figure 2 shows the flow of our system.

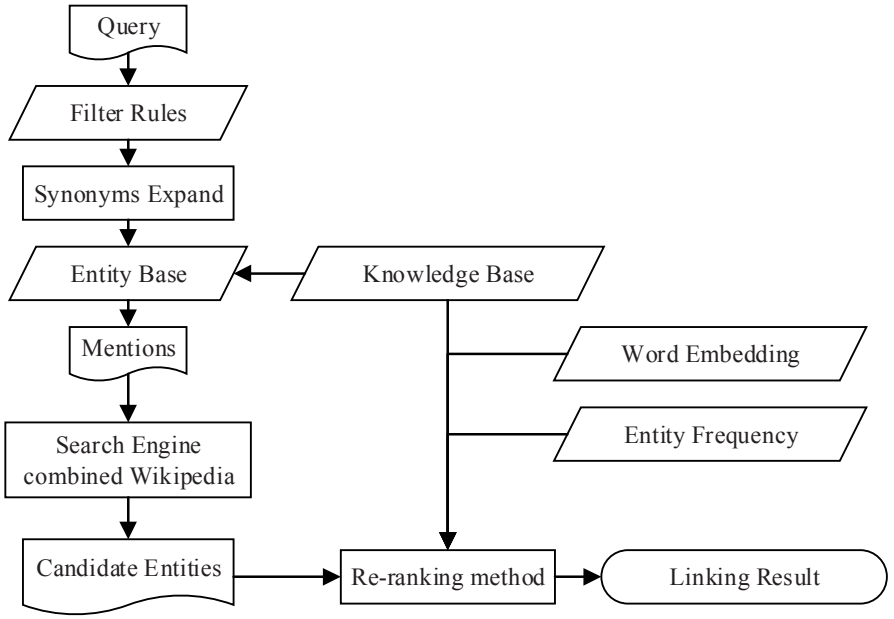

Fig. 2. The workflow of the system

3.1 Pre-processing

We use the Yebol Chinese segmentation system as word segmentation tool. Since we just need to link entities in the knowledge base, so we extract and process the entities, and then build an entity base. Table 1 shows some examples. Synonyms expansion is a simple but effective way to deal with mention variation issue. We use the same method in Meng et al. [4] to build and expand the synonym dictionary. We get "韩国" as alias of "大韩民国", and "海贼王" as a translation of "ONE_PIECE"etc. We select the entity description as feature, and expand the description with Wikipedia and Baidupedia.

Table 1. Examples of entity processing

Original	Result
"爱情公寓1", "爱情公寓2", "爱情公寓3", "爱情公寓4", "爱情公寓_(电视剧)"等	"爱情公寓"
"爱是永恒", "爱是永恒（当所爱是你）"	"爱是永恒"
"茱莉娅·罗伯茨"	"茱莉娅罗伯茨"

3.2 Named Entity Recognition

Traditional entity recognition methods are based on machine learning, which needs plenty of tagged corpus to train a model, and the types of entity in queries are various,

so it's unsuitable to learn a model for entity recognition. In addition, the task only need to link entities in knowledge base, so, to simplify the process and improve the efficiency, we use the following methods to recognize entity.

1. Design refined rules by sample dataset, and then filter the entity mentions
2. Expand entity mentions by synonymy dictionary
3. Match the mentions with entities in entity base: if the mention is complete-matching, we name it identified entity mention M_i, such as "爱情公寓", otherwise, we name it unidentified entity mention M_u, such as "湖人".

3.3 Candidate Entity Generation

Search queries and entity mentions are informal text with noise, for instance, "沙糖桔" is one of misspelling format of "沙糖橘", and there is no matched word in synonym dictionary, so we can't link the entity. Fortunately, search engine provide error correction function and can convert "沙糖桔"to"沙糖橘" before searching. Therefore, we adopt the method based on search engine we used last year [4], we use search engine Baidu, and treat "entity mention" + "中文维基百科"as input query. Figure 3 shows the workflow of entity generation.

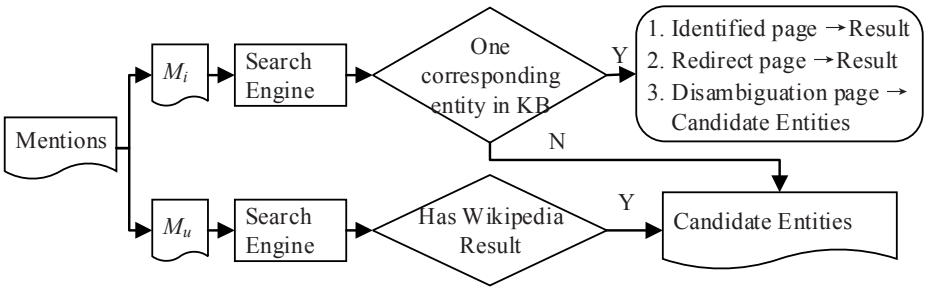

Fig. 3. The workflow of entity generation

If an identified mention has one corresponding entity in the knowledge base, and the first retrieval result is a Wikipedia page, there are three conditions:

1. An identified page, the entity in the title is our linked entity, like "诺基亚".
2. A redirect page, the entity in the title is also our linked entity.
3. A disambiguation page, all the entities in the page are candidate entities, for instance, "Angelababy"and "杨颖（作家）" are candidate entities of mention "杨颖".

If an identified mention has more than one corresponding entities in the knowledge base, all the corresponding entities are candidate entities. For instance, "倚天屠龙记", "倚天屠龙记_(1978年电影)","倚天屠龙记_(1978年电视剧)" etc. are candidate entities of mention "倚天屠龙记". For the unidentified entity mention, if the top 10 pages in the retrieval result contains Chinese Wikipedia pages, the entities in the titles are candidate entities, otherwise, we remove this mention directly.

3.4 Candidate Entity Re-ranking

The most important part of entity disambiguation is to compute the distance between candidate entities' feature and mention's feature. We compute the textual or semantic match-degree between notional words in queries context and candidate entities' description. Mikolov et al. [9] indicated that the word embedding can represent the word in syntax and semantics. So, we can compute the similarity between entity mention and candidate entities by word embedding, and we use cosine similarity. The frequency of entity can tell us the prior probability of the appearance of a candidate entity given the entity mention. Therefore, we propose a re-ranking method combining with match-degree, word embedding similarity and entity frequency to re-rank the candidate entities.

The Match-degree Between Mention Context and Candidate Entity Description
For a set of candidate entities, if there is a candidate entity's score is 1 in method 1, then every candidate entity's score is S_{match1}, otherwise, the score is S_{match2}. We assume c_i is the ith candidate entity, m is the current mention.

Method 1: If the notional word in query appears in the description of a candidate entity, for instance, query: "倚天屠龙记梁朝伟", for entity mention "倚天屠龙记", "梁朝伟" appears in the description of "倚天屠龙记_(1986年电视剧)", it scores 1, otherwise 0.

$$S_{match1}(c_i \mid m) = \begin{cases} 1 & \exists \textit{notional word in query appear}\text{s in the } \textit{description of } c_i \\ 0 & \textit{else} \end{cases} \tag{1}$$

Method 2: Compute the similarity between the search query and entity's description. For the entity's description, we use its notional words' mean vector v_d to represent; for the search query, we use its notional words' mean vector v_q to represent, and the match-degree is the cosine distance between v_d and v_q.

$$S_{match2}(c_i \mid m) = \cos(v_q \ , \ v_d) \tag{2}$$

Therefore, the final score is formula 3.

$$S_{match}(c_i \mid m) = \begin{cases} S_{match1}(c_i \mid m) & \exists i, S_{match1}(c_i \mid m) = 1 \\ S_{match2}(c_i \mid m) & else \end{cases} \tag{3}$$

The Similarity Between Candidate Entity Vector and Entity Mention Vector
We set the candidate entity vector as V_c, and the entity mention vector as V_m, so, the similarity score S_{sim} will be represented as follows,

$$S_{sim}(c_i \mid m) = \begin{cases} 0 & \exists i, v_{c_i} = null \ or \ v_m = null \\ \cos(V_c, V_m) & else \end{cases} \quad (4)$$

Entity Frequency

We use a Chinese Wikipedia corpus to count the frequency of entity, if a candidate entity c_i appears n_{ci} times in the corpus, and entity mention m appears n_m times in the corpus, the frequency score S_{freq} will be:

$$S_{freq}(c_i \mid m) = \frac{2 * n_{ci}}{\sum_{i=1}^{n} n_{ci} + n_m} \quad (5)$$

Hence, the final score of a candidate entity is just like formula 6.

$$S_{final}(c_i \mid m) = \alpha S_{match}(c_i \mid m) + \beta S_{sim}(c_i \mid m) + \gamma S_{freq}(c_i \mid m) \ (i = 1,2,3..n) \quad (6)$$

And α=1, β=0.7, γ=0.2, these parameters are decided by the sample dataset.

We set threshold δ to 0.005, if the difference between highest score and the second highest score is greater than δ, we choose the candidate entity with highest scores as the final linking entity. Otherwise, the second highest candidate entity is also chosen as linking entity. Process in sequence until the adjacent entities' difference is greater than δ, or the linking entity number reaches 5.

4 Experiments

4.1 Dataset

Wikipedia is a high quality encyclopedia containing a wide coverage of named entities, massive knowledge about notable named entities, so it is fit for entity linking work. We download the newest Chinese Wikipedia from wiki dump, and get 707 MB Chinese Wikipedia corpus after processing, which is used to train word embedding and get entity statistical dataset. We use CBOW model [10], [11] in word2vec to train word embedding, and set the dimension to 100.

4.2 Experiment Result and Analysis

Our system score is 61.1% in *Average-F1*, which ranks the second place in fourteen systems. Table 2 shows experiment results. We can see that our system is a little lower than other systems in *Link-Recall*, but our *Link-Precision* is 6.5 percent higher than the third system, and 16.2 percent gap with the first, it shows that our system is promising in entity linking.

Table 2. Part of evaluation results

system	*Link-Precision*	*Link-Recall*	*Link-F1*	*Average-F1*
NO.1	0.724	0.736	0.73	0.733
Ours	0.562	0.695	0.621	0.611
NO.3	0.497	0.704	0.583	0.569

Short context, nonstandard text and various entity representations in search queries make this task difficult. In addition, entity linking task is based on entity recognition, and there will be error in entity recognition inevitably, which may cause error accumulation and pull precision in entity linking down. We concluded that there are mainly three types of error:

1. We adopt a coarse-grained method to recognize entity, for example, there are "北京", "北京交通大学" and "威海" three entities in query "北京交通大学威海校区", and we missed "北京".
2. Some words has no word embedding because of the data sparsity problem. For instance, the candidate entities of mention "天涯明月刀" are "天涯·明月·刀"and "天涯明月刀_(电视剧)" in query "天涯明月刀不删档", while these two candidate entities do not appear in training dataset.
3. The re-ranking method is not precise enough. The linking result of mention "爱情公寓" in query"爱情公寓里的小黑是谁"is "爱情公寓_(电视剧)". However, for "爱情公寓1","爱情公寓2" and "爱情公寓_(电视剧)" etc. Their score of match-degree, similarity and frequency are extraordinary close, so all of them are selected as linking entity.

5 Conclusion

This paper introduces our entity recognition and linking system in search queries. We use a rules based method to recognize entity, then generate candidate entities by search engine and Wikipedia page attributes. Finally, we utilize an entity re-ranking method to score the candidate entities, and get the linking result by the entity score. The results of the experiment shows that our method is effective. In future work, we will optimize the word segmentation result, recognize entity in fine-grained and improve the entity re-ranking method in entity linking.

Acknowledgements. The research work is partially funded by the Natural Science Foundation of China (No.61300081, 61170162), and the Fundamental Research Funds for the Central Universities in BLCU (No. 15YJ03006).

References

1. Mikheev, A., Moens, M., Grover, C.: Named entity recognition without gazetteers. In: Proceedings of the Eacl (1999)
2. Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: Naacl Proceedings of the Conference of the North American Chapter of the Association for Co. (2003)
3. Han, X., Zhao, J.: Nlpr-kbp in tac 2009 kbp track: A two-stage method to entity linking. In: Proceedings of Test Analysis Conference (2009)
4. Meng, Z., Yu, D., Xun, E.: Chinese microblog entity linking system combining wikipedia and search engine retrieval results. In: Zong, C., Nie, J.-Y., Zhao, D., Feng, Y. (eds.) NLPCC 2014. CCIS, vol. 496, pp. 449–458. Springer, Heidelberg (2014)
5. Dalton, J., Dietz, L.: UMass CIIR at TAC KBP 2013 entity linking: query expansion using urban dictionary. In: Text Analysis Conference (2013)
6. Hoffart, J., Yosef, M.A., Bordino, I., et al.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 782−792 (2011)
7. Blanco, R., Ottaviano, G., Meij, E.: Fast and space-efficient entity linking for queries. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 179−188. ACM (2015)
8. Shen, W., Wang, J., Luo, P., et al.: LINDEN: linking named entities with knowledge base via semantic knowledge. In: Proceedings of the 21st International conference on World Wide Web. ACM (2012)
9. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111−3119 (2013)
10. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL (2013)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR Workshop (2013)