Linking Entities in Chinese Queries to Knowledge Graph

Jun Li[1], Jinxian Pan[2], Chen Ye[1], Yong Huang[1],
Danlu Wen[1], and Zhichun Wang[1]*

[1] Beijing Normal University, Beijing, China
[2] Capital Normal University, Beijing, China
zcwang@bnu.edu.cn

Abstract. This paper presents our approach for NLPCC 2015 shared task, Entity Recognition and Linking in Chinese Search Queries. The proposed approach takes a query as input, and generates a ranked mention-entity links as results. It combines several different metrics to evaluate the probability of each entity link, including entity relatedness in the given knowledge graph, document similarity between query and the virtual document of entity in the knowledge graph. In the evaluation, our approach gets 33.2% precision and 65.2% recall, and ranks the 6th among all the 14 teams according to the average F1-measure.

Keywords: Entity Linking, Chinese Query, Knowledge Graph

1 Introduction

Recently, several large scale Knowledge Graphs have been developed [2][1][14][4]. One of the most important applications of knowledge graphs is to enhance web search engines' search result with semantic search information. For example, Google use its knowledge graph to provide structured and detailed information about the search topic in addition to a list of links to websites. Chinese search engines such as Baidu and Sogou also developed their own knowledge graphs and use them for semantic search.

In order to incorporate search engines with knowledge graphs, one important task is to link entities in search queries to knowledge graphs. Recently, much work has been done on the problem of entity linking in documents or tweets. The existing approaches usually use Wikipedia as a knowledge base, identify entities in text and link them to pages in Wikipedia. Only a few work has been done on the problem of entity linking in queries. Radhakrishnan et al. [10] proposed an approach for entity linking for English queries by utilizing Wikipedia inlinks. Blance et al. proposed an approach for fast and space-efficient entity linking for English queries[3]. Entity linking in queries is more difficult than traditional entity linking tasks. First, queries are usually very short texts, it is difficult to find proper context information for disambiguation of entities. Second, there is

* Corresponding author

a very strict time limit for process queries in search engines, so entity linking approaches for queries are supposed to run very efficiently.

In this paper, we report our approach for NLPCC 2015 shared task **Entity Recognition and Linking in Chinese Search Queries**. This task provides a reference Chinese Knowledge Graph and a small size of sample data, which contains several short Chinese queries and the sample results of entity linking. For example, entity linking in a query:"射雕英雄传刘亦菲版" is expected to get the results of linking "射雕英雄传" to a knowledge base entity "pk:tv:射雕英雄传(2006年电视剧)" and "刘亦菲"into a KB entity "pk:per:刘亦菲".

We propose an approach that takes a query as input, and generates a ranked entity links as results. It combines several different metrics to evaluate the probability of each entity link, including entity relatedness in the given knowledge graph, document similarity between query and the virtual document of entity in the knowledge graph. In the evaluation, our approach gets 33.2% precision and 65.2% recall.

The rest of this paper is organized as follows, Section 2 describes the proposed approach in detail; Section 3 presents the evaluation results; Section 4 introduces some related work; Section 5 concludes this work.

2 The Proposed Approach

Our approach first identifies mentions in a given query, and then compute features of each possible mention-entity pairs, based on which the final results are generated.

2.1 Mention Identification

To extract entity mentions in queries, we build a mention dictionary that includes all the entity mentions in Chinese Wikipedia. In Wikipedia, an entity link is annotated by square brackets [[**entity**]] in the source data of articles. Here **entity** denotes the unique name of the referred entity. When the mentioned name of an entity is different from its unique name, the link is annotated by [[**entity** | **mention**]]; **mention** denotes the string tokens that actually appear in the text. In order to get all the mentions that have appeared in Wikipedia, we process all the annotated entity links in the form of [[**entity** | **mention**]] in Wikipedia. In addition, all the titles of articles in Wikipedia are also taken as mentions, which will be included in the mention dictionary. The mention dictionary also records the possible entities that each mention might refer to. Therefore, the dictionary can be represented as 2-tuple $D = (M, E)$, where $M = \{m_1, m_2, ..., m_k\}$ is the set of all mentions in Wikipedia, and $E = \{E_{m_1}, E_{m_2}, ..., E_{m_k}\}$ is the sets of entities corresponding to the mentions in M.

Since we are dealing with Chinese queries, word segmentation tool is used to split queries into lists of terms. Then we match the terms with mentions in the dictionary, if a term precisely matches a mention in the dictionary, we take it as a mention candidate. Each identified mention and its associated entities form a set of mention-entity pairs, which will be scored by several features.

2.2 Features of Mention-Entity Pairs

In order to decide the best entity for each identified mention in the knowledge graph, we propose to use the following features to assess the possibility of the link from a mention to an entity. Given a query q, and a set of mention-entity pairs $P = \{m_i, e_i\}_{i=1}^{n}$, the following features are computed for each mention-entity pair. For a mention-entity pair that has the same name for mention and entity, name length, entity relatedness and document similarity are computed. For other mention-entity pairs, the priori probability, entity relatedness and document similarity are computed. These features are defined as follows.

Name Length

$$f_1(m, e) = \frac{len(e) - minLen}{maxLen - minLen} \tag{1}$$

where $len(e)$ is number of characters in the entity's name. And $maxLen$ and $minLen$ represent the maximum and minimum length of entities in the given knowledge graph. This feature is defined based on the tuition that if a mention match an entity's name, then the longer of the entity's name the more possible the mention refers to the entity.

Priori Probability This feature estimates the probability that a mention m links to an entity e:

$$f_2(m, e) = \frac{count(m, e)}{count(m)} \tag{2}$$

where $count(m, e)$ denotes the number of times that m links to e in the whole Wikipedia, and the $count(m)$ denotes the number of times that m appears in Wikipedia.

Entity Relatedness

$$f_3(m, e, P) = \begin{cases} 1 \text{ if exist an entity in } P \text{ that are linked to } e \\ \quad \text{in the knowlege graph} \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

Document Similarity. We first build virtual document for each entity in the knowledge graph. The virtual document of an entity contains text information of all the other linked entities. The document similarity is calculated between the feature vectors of virtual documents. Before the similarity computation, the virtual document of each entity is represented as a vector, where the elements in the vector are weights assigned to the words in the virtual document using TF-IDF method. For a word i in virtual document j, the weight of the word is computed as

$$\omega_{ij} = tf_{ij} \cdot \lg \frac{N}{df_i} \tag{4}$$

where tf_{ij} is the number of occurrences of i in j, df_i is the number of virtual documents that contain i, and N is the total number of virtual documents. For a entity-mention pair, we compute the document similarity between the query and the virtual document of the entity. The document similarity is computed as the cosine value between their vectors:

$$f_4(m,e,q) = \frac{\sum_{i=1}^{M} \omega_{ie} \cdot \omega_{iq}}{\sqrt{\sum_{i=1}^{M} {\omega_{ie}}^2} \cdot \sqrt{\sum_{k=1}^{M} {\omega_{iq}}^2}} \tag{5}$$

where ω_{ie} and ω_{iq} are the ith weight in the vectors of entity document and query document. M is the total number of distinct words in all of the virtual documents.

2.3 Link Prediction

To predict links from mentions to entities in the knowledge graph, our approach computes the weighted sum of features between mentions and entities by the following score functions:

$$S_1(m,e,q,P) = \omega_1 \times f_1(m,e) + \omega_3 \times f_3(m,e,P) + \omega_4 \times f_4(m,e,q) \tag{6}$$

$$S_2(m,e,q,P) = \omega_2 \times f_2(m,e) + \omega_3 \times f_3(m,e,P) + \omega_4 \times f_4(m,e,q) \tag{7}$$

S_1 is for the mention-entity pairs that have the same names; S_2 is for the mention-entity pairs that have different names. All the candidate mention-entity pairs generated from one query are ranked by their score in descending order. And the top-k mention-entity pairs whose scores are larger than a threshold δ are the results of entity linking in the query.

In our system, all the parameters are set empirically. Their values are listed in Table 1.

Table 1. Setting of Parameters

Parameter	**Value**
ω_1	0.2
ω_2	0.2
ω_3	0.4
ω_4	0.4
k	3
δ	0.3

3 Evaluation Result

In this section, we will first introduce the evaluation dataset and the evaluation metrics, and then present the evaluation results of our approach.

3.1 Dataset

NLPCC 2015 shared task **Entity Recognition and Linking in Chinese Search Queries**, a file of the knowledge graph is provided by the organizer. This file contains large number of entities and relations between them. Each line represents a record in the knowledge base, and each record has 6 columns: subject ID, predicate ID, object ID, subject, predicate, object. A subject means a entity and the subject ID means the entity ID. As for the task requirement, we should link every named entity into the knowledge for each short query. And each query consists of an ID and a short search query. The answer we give should be in from of a query ID followed by an entity ID.

3.2 Evaluation Metrics

For a given query, we evaluate system performances using average F1-scores. Given a query q, the output of a system S^*, containing $|S^*|$ different groups of linking result for the named entities appearing in the given query. We compute the precision, recall and F1-score by comparing S^* with the answer the competition organizer providing S.

$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8}$$

The final average F1-score is obtained by averaging over each query.

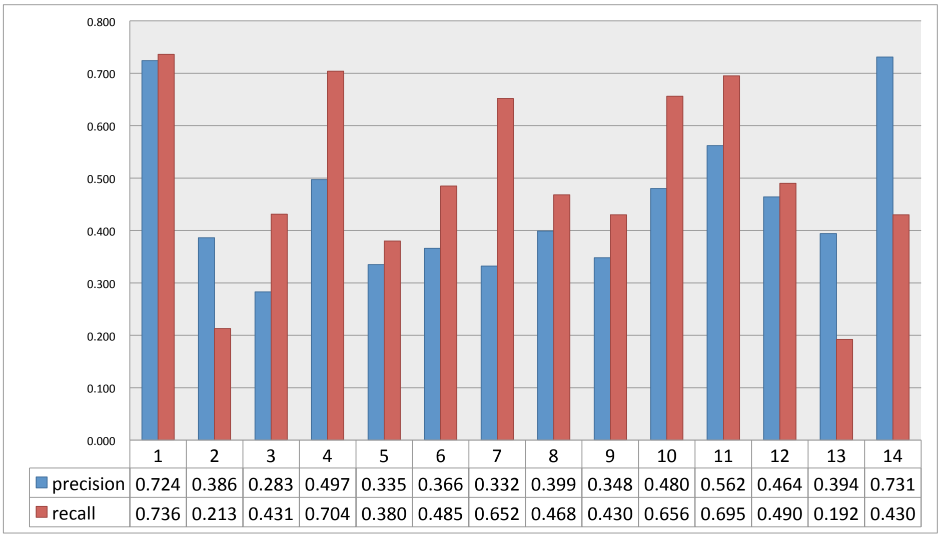
3.3 Results

Fig. 1 shows the precision and recall of each participant system. Team 7 is our team ID. Our system gets 33.2% precision and 65.2% recall. The precision of our system is not satisfied while the recall is not very low. We think the low precision is due to the manually set parameters, they might not be the optimal values. If we use the Machine Learning method to set the weight of each feature, we may get a better result. We think low precision also results from our mention identification method. Because our goal is to identify as much as mentions in the mention identification procedure, it also bring much noise which influence our precision at last. All these problems will be investigated and hopefully settled in the future. Fig. 2 shows the average F1 score of each team, our team gets a medium rank (the 6th) among all the 14 teams.

4 Related Work

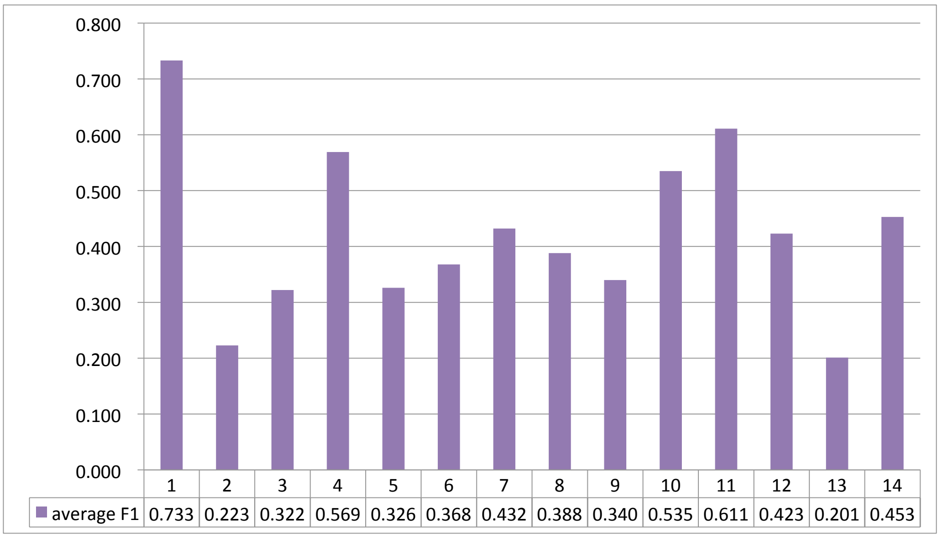
In this section, we review some related work. Lots of work has been done in the problem of *Entity Linking*, which aims to identify entities in documents and link them to a knowledge base, such as Wikipedia and DBpedia.

Wikify! [8] is a system which is able to automatically perform the annotation task following the Wikipedia guidelines. Wikify! first uses a unsupervised



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
🟦 precision	0.724	0.386	0.283	0.497	0.335	0.366	0.332	0.399	0.348	0.480	0.562	0.464	0.394	0.731
🟥 recall	0.736	0.213	0.431	0.704	0.380	0.485	0.652	0.468	0.430	0.656	0.695	0.490	0.192	0.430

Fig. 1. Evaluation results: precision and recall



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
🟪 average F1	0.733	0.223	0.322	0.569	0.326	0.368	0.432	0.388	0.340	0.535	0.611	0.423	0.201	0.453

Fig. 2. Evaluation results: F1-measure

keyword extraction algorithm to identify and rank mentions; and then it combines both knowledge-based approach and data-driven method to predict the links from mentions to entities in Wikipedia. Milne et al. [9] proposed a learning based approach for linking entities in text to Wikipedia. Their approach trains a C4.5 classifier based on three features of entity-mention pairs for link disambiguation. Kaulkarni et al. [6] proposed a collective approach for annotating Wikipedia entities in Web text. Their approach combines both local mention to entity compatibility and global document level topical coherence. The collective prediction of entity links improves the accuracy of results. Other collective entity linking approaches include [5][12][11].

The above entity linking approaches mainly handle long documents, there are also some work on linking entities in tweets to knowledge graphs [13][7]. To perform entity linking in short tweets, these approaches usually use users' other information to help disambiguate entities in tweets, such as current user's other tweets or current user's social network information. But in the NLPCC 2015 shared task, there is no other associated information of queries, so it is more difficult to identify and link entities in queries. There are several approaches for entity linking in English queries. For example, Radhakrishnan et al. [10] proposed an approach for entity linking for English queries by utilizing Wikipedia inlinks; Blance et al. [3] proposed an approach for fast and space-efficient entity linking for English queries.

5 Conclusion and Future Work

In this paper, we report technique details of our approach for NLPCC 2015 shared task **Entity Recognition and Linking in Chinese Search Queries**. Our approach takes a query as input, then first generates a set of candidate mention-entity pairs from the query; four features are proposed to evaluate the possibility of each mention-entity pair; an aggregated score is computed for each candidate, based on which the final entity linking results are drawn.

Our approach gets 33.2% precision and 65.2% recall, and ranks the 6th among all the 14 participant teams by average F1-measure. The future work includes defining new features weighting methods to further improve the results of our approach. And for the task of entity linking in queries, running time is also a very important factor we should consider. So we will also test and improve the efficiency of our approach.

6 Acknowledgement

The work is supported by NSFC (No. 61202246), NSFC-ANR(No. 61261130588), and the Fundamental Research Funds for the Central Universities (2013NT56).

References

1. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th international semantic web conference*, pages 722–735, 2007.
2. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009.
3. R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 179–188, New York, NY, USA, 2015. ACM.
4. K. D. Bollacker, R. P. Cook, and P. Tufts. Freebase: a shared database of structured general human knowledge. In *Proceedings of the 22nd national conference on Artificial intelligence*, volume 2, pages 1962–1963, 2007.
5. X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774, 2011.
6. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466, 2009.
7. X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, 2013.
8. R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, 2007.
9. D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, 2008.
10. P. Radhakrishnan, R. Bansal, M. Gupta, and V. Varma. Exploiting wikipedia inlinks for linking entities in queries. In *Proceedings of the First International Workshop on Entity Recognition &#38; Disambiguation*, ERD '14, pages 101–104, New York, NY, USA, 2014. ACM.
11. W. Shen, J. Wang, P. Luo, and M. Wang. LIEGE: link entities in web lists with knowledge base. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1424–1432, 2012.
12. W. Shen, J. Wang, P. Luo, and M. Wang. LINDEN: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st international conference on World Wide Web*, pages 449–458, 2012.
13. W. Shen, J. Wang, P. Luo, and M. Wang. 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*, KDD '13, pages 68–76, New York, NY, USA, 2013. ACM.
14. F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.