HIT Approaches to Entity Linking at TAC 2012

Yuhang Guo, Bing Qin, Ting Liu*, Sheng Li
Research Center for Social Computing and Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology, China
{yhguo, bqin, tliu*, sli}@ir.hit.edu.cn

Abstract

This paper describes the system of HIT at the 2011 Text Analysis Conference (TAC) Knowledge Base Population (KBP) track English Entity Linking task. In this task, a system is required to link a name string in a given contextual document to its referent entity in an external knowledge base. The HIT system makes the linkage by using the relatedness score between the context and the model of the entity. In this system, an entity is modeled based on its information from Wikipedia. The relatedness score contains three parts: the popularity of the entity along with the query name, the context language model score which is generated by the entity model and the occurrence of the aliases of the entity. The evaluation result shows the system performance is comparable with the median performance of the participating systems.

1 Introduction

Research Center for Social Computing and Information Retrieval from Harbin Institute of Technology (HIT) participated in the Entity Linking task at the 2012 Text Analysis Conference (TAC) Knowledge Base Population (KBP) track. This paper describes the system we implemented.

Entity Linking is the task of linking a name mention in a document to the correspondence entity in a Knowledge Base (KB) (McNamee and Dang, 2009; Ji and Grishman, 2011). In the TAC-KBP track, the input of the entity linking is comprised of a KB and a query which contains a name string and the source document which the string appears in. The KB of 818,741 entities in this track is a subset of the Wikipedia entity collection. The output of the task is the correspondence entity id in the KB for the input query. If the entity is out of the KB, the system returns a unique NIL id of this entity.

2 Our Approach

We resolve the entity linking problem in three steps. First, we generate a set of candidate entities for each query name. Then we rank these candidates according to the similarities between the candidate and the query context in the source document. Finally, we discriminate those queries of out-of-KB entities.

2.1 Candidate Generation

In this step, we aim to collect all potential entities of the query name. Probably the most direct way is to retrieve the query name in the Wikipedia, and then harvest the entity with the name. However, many complex cases make this step need more sophisticated processing.

Some query names cannot be directly found in the Wikipedia not because the corresponding entities are not in it but because the query names are aliases or alternative names of the entities which are not included in the name field of the relevant pages. Wikipedia provides a redirect mechanism to link popular aliases or synonyms to the corresponding pages. For example, the page titled with *Robert Gates* could be found through the redirect page of the alias *Bob Gates*.

Redirect pages cover most popular aliases. However there are still many names which could not be recalled in that way. We mine other aliases from following sources and map them to the corresponding entities. Here we use the term "alias" to represent all other names of the entity except for the article title of the entity.

In some Wikipedia articles, structured information is organized with **Infobox** template in attribute-value pair format. We extract the values of the attribute "fullname" or "nickname" in the Infobox to supplement the alias set of the entity of this article.

In the first paragraph of Wikipedia article, the name/names of the entity this article describes is/are usually highlighted in bold format. So we extract these **bold texts** as the aliases of the entity.

Wikipedia contains plenty of cross references in the form of hyperlink. The **hyperlink anchor texts** can be different from the name of the target pages. We collect these anchor texts as the aliases of the corresponding target entities.

In Wikipedia, if a name is shared by several entities these entities are usually listed in a **disambiguation page** of this name. We augment the alias set for each listed entities with this name (after removing the *(disambiguation)* suffix if it contains).

In all, we have extracted 23,895,597 name-entity pairs including 19,115,923 names and 4,068,377 entities from Wikipedia.

In our system, we employed an open-source Java-based Wikipedia API (Zesch et al., 2008) to extract the Wikipedia texts.

For the acronym query names, we try to find its full name coreference in the source document with patterns:

If the acronym is bracketed, we extract the name phrase immediately before the capitalized letter nearby (e.g. ... *The Mexican Football Federation (FMF) on Monday* ...).

If the acronym is followed by a bracket, we extract the phrase in the bracket (e.g. ... *From the PRC (People's Republic of China) we get much benefit.* ...).

Or else, we just find the phrase in the context with the same capitalized letter as the acronym (e.g. *ABC* → ... *he told the Australian Broadcasting Corporation.* ...).

When the full name is found, we use this full name to generate the candidates instead of the original query name.

Similar to the acronym extension, we also attempt to extend non-acronym query names to their longer forms. We first extract named entities around the query name using a standard NER tool (Finkel et al., 2005). And then we extract the first named entity string which contains the query name as the extension.

If the query extension is included in the name list which we have extracted from Wikipedia, then the query name is substituted by the extension. Other wise the system use the original query name to generate the candidates.

2.2 Candidate Ranking

After the candidate generation step, nearly every query got more than one candidates. In the candidate ranking step, we need to identify which candidate is most likely to be the referent entity of the query. We rank the candidates by their relatedness to the query.

The relatedness score includes three parts: the probability of the query name and the candidate entity, the co-occurrence probability of the candidate entity and all its names in the query document given the entity, and the language model score of the context which is generated by the candidate entity model.

Intuitively, a popular entity in Wikipedia will also appear in document in a high probability. On the other hand, the probability of an entity appears in document is different given different names. We use the co-occurrence frequency of the query name and the candidate entity to estimate the joint probability of the name and the entity in the document. The co-occurrence frequency is mainly counted from the anchor text and the target entity of hyperlinks in Wikipedia.

Furthermore, if several aliases of the candidate appear in the query document, our confidence on this candidate should be enhanced. The system sum up all the alias-entity probability given the candidate as a second relatedness score. The alias-entity co-occurrence probability is calculated as the query name and candidate entity co-occurrence probability.

In our system, the contextual relatedness of the candidate entity and the query is calculated under the language model framework. For each candidate entity, we train its language model based on the entity related text from Wikipedia. The text includes the

Runs	All	in KB	not in KB
Highest	0.730	0.687	0.847
Median	0.536	0.496	0.594
HIT	0.525	0.519	0.532

Table 1: The highest, median and our entity linking results

Wikipedia page of the candidate entity and all the pages which contains a link to this page. The text is segmented into name strings which is included from the name list we have extracted from Wikipedia. We extract the query context around the query name in window size 50 from the query document. The Dirichlet prior method is used for the model smoothing and the smoothing parameter is set according to our experiment on TAC-KBP 2009 and TAC-KBP 2010 data sets.

2.3 NIL Labeling

Some of the entities for the query are out of the track KB. In TAC-KBP 2012, queries need to be clustered and labeled with KB ID or NIL ID. The NIL ID should start with "NIL" and be suffixed with an identifier of the cluster.

We implemented a simple labeling method based on following rules:

If the score of the top candidate is higher than the threshold and the top candidate is in the KB, then label the query with the KB ID of the candidate.

If the query is an acronym, then suffixes the NIL mark with the full form of the acronym.

Or else if the candidate set is empty, then suffixes NIL with the query name.

Otherwise suffixes NIL with the name of the top score candidate.

3 Results

In the TAC-KBP 2012 Entity Linking track, systems are evaluated by B-Cubed+ precision, recall and F1 score[1] (F1 score is the official score). 25 teams submitted 98 runs in total. The results of the highest, median and our run are listed in Table 1.

4 Conclusion

In this paper, we describe our entity linking system for TAC-KBP 2012. The system For the NIL processing we leverage a simple heuristic method. Evaluation results show that the system performance is comparable to median in the track.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 60975055, 61073126, 61133012 Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2009069), and Fundamental Research Funds for the Central Universities (HIT.KLOF.2010064).

References

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA, June. Association for Computational Linguistics.

P. McNamee and H.T. Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Proceedings of the Second Text Analysis Conference (TAC2009)*.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

[1] See http://nlp.cs.qc.cuny.edu/kbp/2011/scoring.html for the metric details.