A Multiple-Stage Framework for Related Entity Finding: FDWIM at TREC 2010 Entity Track

Dong Wang, Qing Wu, Haiguang Chen, Junyu Niu
Department of Computer Science and Technology
Fudan University, Shanghai, China
{wangdong,qingwu,hgchen,jyniu}@fudan.edu.cn

Abstract

This paper describes a multiple-stage retrieval framework for the task of related entity finding on TREC 2010 Entity Track. In the document retrieval stage, search engine is used to improve the retrieval accuracy. In the entity extraction and filtering stage, we extract entity with NER tools, Wikipedia and text pattern recognition. Then stoplist and other rules are employed to filter entity. Deep mining of the authority pages is proved to be effective in this stage. In entity ranking stage, many factors including keywords from narrative, page rank, combined results of corpus-based association rules and search engine are considered. In the final stage, an improved feature-based algorithm is proposed for the entity homepage detection.

1 INTRODUCTION

With the rapid development of World Wide Web, web data is a huge wealth for many fields. In the past, users will get a lot of documents when they want to find some information. There is still a lot of work to do if users want to find exact information, for example exact entities or their homepage. In entity track of TREC 2010, the main task is related entity finding, which is defined as follows:

> *Given an input entity, by its name and homepage, the type of the target entity, as well as the nature of their relation, described in free text, find related entities that are of target type, standing in the required relation to the input entity.*

To finish this task, we design a multiple-stage framework which consists of the following five stages: document retrieval, entity extraction, entity filtering, entity ranking, and homepage detection. We raise some new conceptions such as authority pages and employ some methods including the combination of corpus-based association rules and search engine.

2 APPROACHES

2.1 Document Retrieval

For the whole framework, the initial stage is document retrieval. We consider the entities extracted from those authority pages to be more reliable. We do parsing for the narrative, employ stoplist,

extract keywords, and finally generate query. These queries are put into Google and the top 10 return pages in the clueweb09 are selected as our document collection. The document weight is designed according to Google's rank, which will be explained in detail in the entity ranking section. We also add authority pages to the document collection, of which we do deep mining. In our experiments, authority pages are defined as source entity homepages given by the query and their Wikipedia pages. We extract entities from tables and list of the authority pages using both text content and DOM tree, and then give these entities higher score in the entity ranking stage.

2.2 Entity Extraction

Entity Extraction is the second stage of the framework. We mainly use Stanford Named Entity Recognition[1] to extract entity. Also we adopt text pattern recognition methods to improve the accuracy of the entity extraction.

2.3 Entity Filtering

Entity filtering is the third stage in the framework. It mainly depends on stoplist. Other rules include the length of the entity and whether the entity is a Wikipedia title. We filter each kind of entity with a corresponding upper and lower limit length. After entity filtering, we get the candidate entity set *E*.

2.4 Entity Ranking

Entity Ranking is the fourth stage of the framework. We focus on how to measure the accuracy of those candidate entities. For a candidate entity *e* in the set *E*, its score is computed as follows:

2.4.1 Entity Frequency

Entity frequency in the document collection *D* is calculated by:

$$\boldsymbol{f_{fre} = \frac{N_e - N_{min}}{N_{max} - N_{min}}(f_{fre\_max} - f_{fre\_min})}$$

where $\boldsymbol{N_{max}}$ is the maximum frequency of the entity extracted from the document collection, and $\boldsymbol{N_{min}}$ is the minimum, $\boldsymbol{N_e}$ is the frequency of entity *e*, $\boldsymbol{f_{fre\_max}}$ is the score of the entity which has the maximum frequency (usually 1.0 for normalized), and $\boldsymbol{f_{fre\_min}}$ is the minimum. Besides, entities extracted from authority page are counted twice.

2.4.2 Document Ranking

As we use Google's return pages as our document collection, page rank by Google can be a good reference. We rank the document according to f_{page} calculated by

$$\boldsymbol{f_{page} = \alpha\frac{(P_{all} - P_e + 1)}{P_{all}} + (1 - \alpha)}$$

where α is the weight parameter between 0 and 1. P_{all} is the page number(in the experiment,

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

$P_{all} = 10$), and P_e is the rank of page *e* extracted from. For multiply pages, we use the top one.

2.4.3 Confidence of *e* to Source Entity

We combine corpus-based association rules and search engine to calculate the relationship between *e* and source entity. This method is originally proposed by Richard Chow for privacy leak detecting[1]. The confidence of an inference *e* to source can be computed by:

$$\boldsymbol{f_{co-occ} = \frac{Res(Source \,\&\, Candidate\ Entity)}{Res(Candidate\ Entity)}}$$

where $Res(Source \,\&\, Candidate\ Entity)$ is the result number returned by search engine for the query source entity and candidate entity , and $Res(Candidate\ Entity)$ is for candidate entity only. E.g., for the source entity "*Microsoft*" and a candidate entity *"Windows 7"*, Google return 74,700,000 results for *"Microsoft Window 7",* and 483,000,000 results for *"Windows 7",* so the confidence of an inference "*Windows 7*" to "*Microsoft*" is $74700000/483000000 \approx 0.155$.

2.4.4 Keywords

To check the consistency between *e* and query narrative, we employ keyword. This is similar to [2]. They define the first term or phrase of each query narrative as the only keyword, and in our system, we use multiple keywords. The keywords are extracted from the query narrative, which can be plural noun, organization, location, date and other term or phrase. For example , "Authors awarded an Anthony Award at Bouchercon in 2007" , the keywords can be "Authors" , "Anthony Award", " Bouchercon", "2007". We parse the query narrative with WordNet and other method. It is similar to do an entity extraction to generate query narrative, but not the same (In run FDwimET3, we extract keywords manually instead of using WordNet). Due to the variety of keywords, we believe that the frequency or category can not reflect the confidence well, so we only use the distance. If one keyword repeats for several times in the document, we use the one nearest to *e*. The minimum distance between the entity and keyword is f_{key} calculated by:

$$\boldsymbol{f_{key} = \beta * min\{\frac{D(e, keyword)}{L_{doc}}\} + (1 - \beta)}$$

where β is the weight parameter between 0 and 1, $D(e, keyword)$ is the distance between *e* and keyword, L_{doc} the length of document, $doc \in D$, the document collection. If keyword and *e* do not co-occurrence at any document, then $f_{key} = 0$. We use two keywords in our experiments, the results are marked as f_{key1} and f_{key2} separately.

2.4.5 Final Ranking

Consider the above factors, we use a linear combination of each score to yield the ranking score f_e' for *e* as follow:

$$\boldsymbol{f_e' = \varepsilon_1 f_{fre} + \varepsilon_2 f_{page} + \varepsilon_3 f_{co-occ} + \varepsilon_4 f_{key1} + \varepsilon_5 f_{key2}}$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5$ are the combination coefficients between 0 and 1 for the scores f_{fre}, f_{page}, f_{co-occ}, f_{key1}, f_{key2}, and $\sum_{i=1}^{5} \varepsilon_i = 1$.

For entities extracted by text pattern recognition method described in section 2.2,

$$\boldsymbol{f_e = f_e' + (1 - f_e')/C}$$

where C is a constant greater than 1. For other entities,

$$\boldsymbol{f_e = f_e'}$$

f_e is the final ranking score for *e*. Then we rank the entity list according to f_e.

2.5 Homepage Detection

In the final stage, an improved feature-based algorithm is proposed for the entity homepage detection, in which features includes URL features, page content features and others. For every entity in the entity list, we just use its name as a query, and get Google's top 5 return non-Wikipedia pages that are in the clueweb09 as its candidate homepages. Then each page can get a score by the feature-based algorithm. The highest scoring page is selected as the homepage of the entity.

3 RESULTS

We submitted the following four runs:
FdWimET1: Run the complete process described above;
FdWimET2: Run with different value of combination coefficients in Section 2.4.5 and fewer filtering rules compared to FdWimET1
FdWimET3: Run with keyword selected manually described in Section 2.4.4
FdWimET4: Run without using Wikipedia in the stage of entity filtering.

Runs	P@10	nDCG@R	Map	R-prec	Rel_ret	Pri_ret
FdWimET1	**0.3234**	0.3259	0.2235	0.2823	83	276
FdWimET2	0.3170	0.3382	**0.2272**	**0.2917**	120	303
FdWimET3	0.3213	0.3376	0.2218	0.2886	116	297
FdWimET4	0.3128	**0.3420**	0.2223	0.2837	**140**	**333**

Table1: Performance of all submitted runs for P@10 ,NGCG@R, Map, R-prec scores, and the number of relevant and primary entities (related homepage returned and primary homepage returned, respectively) retrieved

Table 1 lists the results for our four runs. From the table, we can see that all P@10 and nDCG@R scores are over 0.3, which proves that deep mining of authority pages and employing text pattern recognition in the entity extraction stage is effective for related entity finding. For FdWimET1 vs FdWimET2, we can see that strict filtering rules result in fewer return pages but higher P@10 score. For FdWimET4 in which Wikipedia is not used for filtering entities, more entities are extracted compared to other runs. That is the reason why FdWimET4 returns the most primary and related homepages.
Figure 1 shows the NDCG@R scores of our best run and best run of all TREC2010 runs for each topic. For topics 35, 46, and 59 are not judged official, so the evaluation results are over 47 topics. According to the figure, our system returns highest NDCG@R scores for 8 topics. Also there are 6 out of 47 queries got zero. The problem might be in the document retrieval stage and/or the entity extraction stage(depending too much on NER tools).

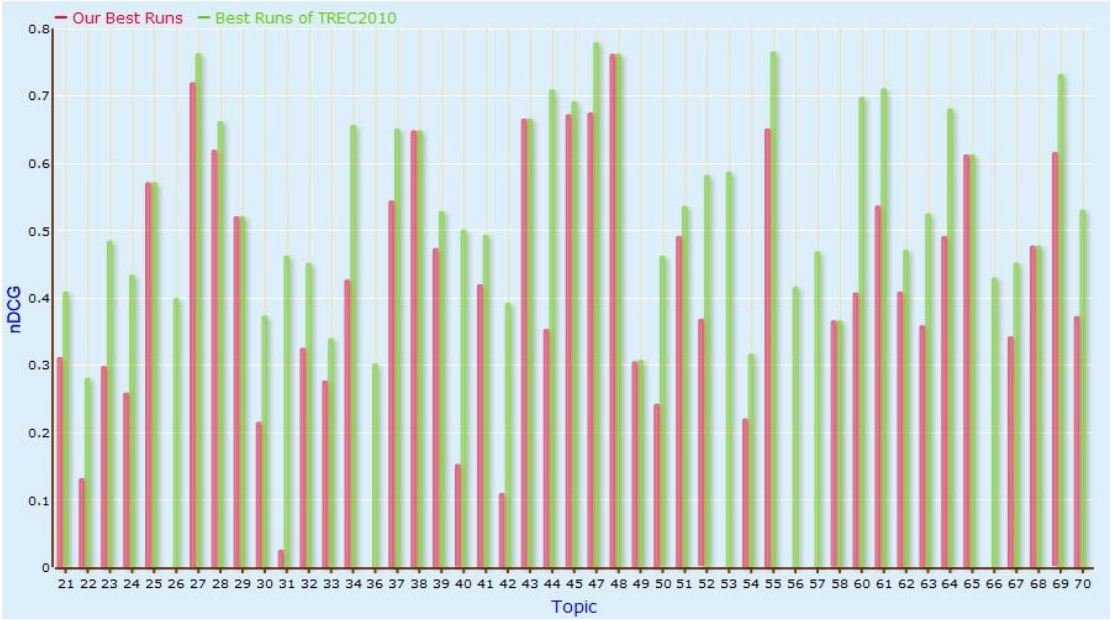

Figure 1: NDCG@R scores of our best run and best runs of TREC2010 for each topic

4 CONCLUSIONS

In this paper, we describe our multiple-stage retrieval framework for the task of related entity finding. After document retrieval stage and entity extraction and filtering stage, candidate entity set is generated. Search engine, NER tools, Wikipedia, text pattern recognition, stoplist and other tools or methods are employed in these stages. Then many factors including keywords from narrative, page rank, combined results of corpus-based association rules and search engine are considered for ranking entities. Finally, an improved feature-based algorithm is proposed for the entity homepage detection. According to the results of our submitted runs, we confirm that deep mining of authority pages and employing text pattern recognition in the entity extraction stage is effective for related entity finding.

In future work we will focus on: definition and extended use of authority pages; entity extraction without using NER tools; other factors which can reflect the relation between query and candidate entity; continued research on the feature-based homepage detection algorithm.

5 ACKNOWLEDGMENTS

We would like to sincerely thank Lin Bao and Jiachen Chen. In addition, our work is supported by 863 Program of China 2009AA01Z429.

6 REFERENCES

[1] Chow, R. ; Golle, P. ; Staddon, J. Detecting privacy leaks using corpus-based association rules. In *KDD*, Las Vegas, NV, 2008

[2] Y. Fang, L. Si, Z. Yu, Y. Xian, and Y. Xu. Entity Retrieval with Hierarchical Relevance Model, Exploiting the Structure of Tables and Learning Homepage Classifiers. In *TREC* 2009, Gaithersburg, MD, 2009.