# PRIS at TREC 2010: Related Entity Finding Task of Entity Track

Zhanyi Wang, Chunsong Tang, Xueji Sun, Haoyi Ouyang,
Ru Lan, Weiran Xu, Guang Chen, Jun Guo
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China
wangzhanyi@gmail.com

**Abstract**. This paper reports the approaches to the task of Entity Track applied by PRIS lab of BUPT in TREC 2010. We used Document-Centered Model (DCM) and Entity-Centered Model (ECM) for the task. BM25 method was introduced into ECM besides indri retrieval model. Another improvement aimed at entity extraction. Special web page, NER tool and entity list generated by some rules were all taken into account. Also, some external resources such as Google and CMU search engine were applied.

## 1. Introduction

The overall aim of the 2010 TREC Entity track is to create a test collection for the evaluation of entity related searches on Web data. This year the Related Entity Finding (REF) is the main task of the track. The problem of related entity finding is defined as follows [1]: 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. Compared with previous edition, following changes are introduced.

- English subset of ClueWeb cat A
- Single record submission format
- No supporting documents
- New entity type: location
- Revised definition of primary and relevant homepages
- Wikipedia pages are not accepted
- Primary homepages are rewarded more
- Names are judged only for primary pages; the judgment is binary

In order to cope with the challenges, we mainly aimed at four areas: collecting relative

documents, extracting named entities, constructing retrieval model and allocating homepages for entities. We also used some external resources, such as Stanford NER Tool, Indri[1] & Lemur toolkit[2] and Google and CMU search engine.

The report is organized as follows. Section 2 describes the process of collecting relative documents. Section 3 introduces our methods of named entity extraction. Section 4 proposes our retrieval models. Allocating homepages for entities is presented in Section 5. Submitted runs show in Section 6 and Section 7 gives the conclusion and future work.

## 2. Collecting Relative Documents

The most significant change in TREC 2010 Entity Track is the enlargement of data set. The volume of English portion of uncompressed ClueWeb category A is about 15 TB. Being restricted by hardware resource, we used the category A search engine developed by CMU (http://boston.lti.cs.cmu.edu:8085/ clueweb09/search/cata_english/lemur.cgi?). We collected documents related to queries for named entity finding.

First, we analyzed all 70 topics with focus on the “entity_name” and “narrative” fields, extracted keywords from these fields and reformulated queries. Taking the 30th topic for example, the original topic is as the follows:

```
<query>
<num>30</num>
<entity_name>Ocean Spray Cranberries, Inc.</entity_name>
<entity_URL>clueweb09-en0132-45-30062</entity_URL>
<target_entity>location</target_entity>
<narrative>Find U.S. states and Canadian provinces where Ocean Spray growers are located.
</narrative>
</query>
```

Figure 1. An example of a topic.

In this topic, “Ocean”, “Spray” and “grower” can be treated as keywords. Then we rewrote it into the type of Indri query language, such as “#uw(Ocean+Spray+grower)”. The actual query language was more complicated.

Second, we sent the queries to the CMU search engine one by one. Some useful information

[1] http://www.lemurproject.org/indri.php
[2] http://www.lemurproject.org/

such as document number, title, URL, rank and score could be got from the result page. Generally, a query obtained less than 1,000 documents. This information was stored for the latter needs.

In order to indexing by Indri or Lemur, we parsed the documents and structure into the follow format:

```
<DOC>
<DOCNO>clueweb09-en0003-81-01650</DOCNO>
<DOCHDR>
URL: http://shop.crackberry.com/content/smartphones/index.htm
</DOCHDR>
<TITLE>Smartphone Connections - BlackBerry</TITLE>
<TEXT>
Document content
</TEXT>
</DOC>
```

Figure 2. An example of a document.

## 3. Named Entities Extraction

Last year, named entities extraction in our system only adopted automatic NER tool. The precision was low. In contrast, we integrated multiple methods this year. Specifically, special web page, NER tool and entity list generated by some rules were taken into account.

First we refined key words from topics and retrieved in Google and Wikipedia. In related pages with high ranking score, we looked for special tables and lists, and then screened partial entities for the topic manually.

Then we extracted entities using NER tools from the web pages. We filtered the entities through several entity lexicons to gain the ultimate list. With the development of NLP, most NERs (Named Entity Recognizers) can utilize the context information to recognize the named entities, such as Stanford NER[3], LBJ NER[4]. After weighting the speed and accuracy of Stanford NER and LBJ NER, we decided to employ the former.

In the previous section, we gained the score, doc-number and URL of the pages with high relevance retrieved by searching engine. Then the first five pages were crawled and downloaded to the local. Then we picked out the related named entities in the simply-processed documents

[3] http://nlp.stanford.edu/ner/index.shtml
[4] http://cogcomp.cs.illinois.edu/page/software_view/4

using the Stanford NER. Certainly, the NER can identify all types of entities at one time, which means we should limit the condition to get the entities of target type. Finally, a list of entities in a particular format was gained under the following procedures.

However, not all the results extracted were entities, so we built lexicons using the Wikipedia resources to refine the list to gain a better result.

Wikipedia is kind of an encyclopedia including human knowledge in all fields, rather than a simple dictionary, an online forum or others. It is worth mentioning that Wikipedia is an open resource to the extent that anyone can copy and modify the materials. It offers convenience for people of different occupations to access to knowledge. On the other hand, users can enrich them by broadening their scope of knowledge.

Considering above features of Wikipedia, we downloaded and stored the pages as local texts. Meanwhile, we discriminated the texts by rules, such as, starting with "Organization", starting with "Companies" for ORG type. We referred to the rules worked out by University of Amsterdam last year [2] and proposed some new ones. Then we obtained four different collections of documents including the four types.

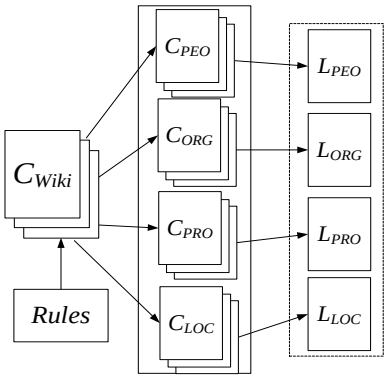


Figure 3. Build entity lists of four types.

Figure 3 shows the whole course. In each collection, we extracted a list of entities according to the labels in the documents, which is called lexicon. After the NER tool's result, we just need to justify whether the entity is in the corresponding sort of lexicon or not.

## 4. Retrieval Models

This year, we submitted four runs. The first two runs (PRIS1 and PRIS2) used Document-Centered Model (DCM) which is similar to the two-stage search model [3] we proposed last year. As figure 4 shown, we call it DCM for emphasizing that documents link the

queries and entities, as opposed to Entity-Centered Model (ECM). The difference is that entities extracted from special pages are given more weights in PRIS2. So the more reliable entities are, the early ranked they are. The query, entity, document and collection are denoted by *q*, *e*, *d* and *D* respectively. Then the formula of DCM is

$$score(e,q) = p(e|q) = \frac{p(q,e)}{p(q)} = \frac{1}{p(q)} \sum_{d \in D} p(e|d)\, p(q|d)\, p(d) \tag{1}$$

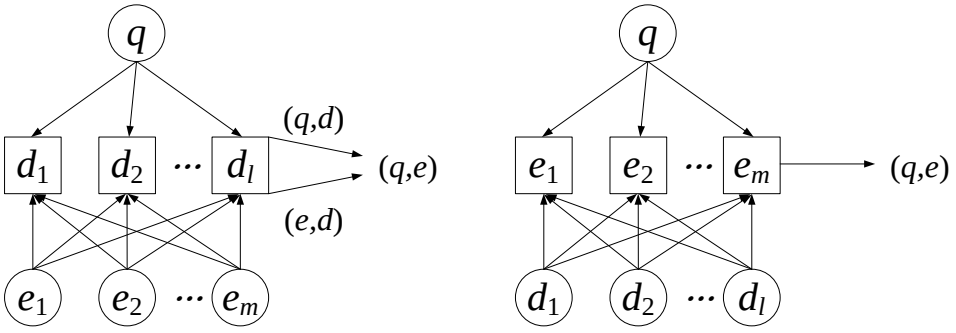


Figure 4. The framework of DCM and ECM.

A new model was introduced in our task this year. In ECM, an entity is represented by snippets extracted from relevant documents. Unlike DCM, both probabilities of a query and entity with respect to a document are estimated by twice retrievals, the probability of an entity given by a query is got only once retrieval in ECM. The formula of ECM is

$$score(e,q) = p(e|q) = p(d_e|q) = \frac{p(q|d_e)\, p(d_e)}{p(q)} \tag{2}$$

where $d_e$ is the new document composed of snippets of an entity *e*.

As is well known, the context of an entity in a document is considered to be the significant information. Words around it are called a snippet. In our task, the length of every snippet was 150 words. Some snippets extracted from different relevant documents of the entity were combined into a new document. Each new document contains original entity information. We built an index to the new documents for every topic. As long as the document is retrieved by a query, the entity is considered to be relevant. In ECM, we used the BM25 weight to rank documents.

BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document. One of the most prominent instantiations of the function is as follows.[5]

Given a query *Q*, containing keywords $q_1,...,q_n$, the BM25 score of a document *D* is:

[5] http://en.wikipedia.org/wiki/Okapi_BM25

$$score(D,Q)=\sum_{i=1}^{n} IDF(q_i)\cdot\frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)} \quad (3)$$

where $f(q_i,D)$ is $q_i$'s term frequency in the document $D$, $|D|$ is the length of the document $D$ in words, and *avgdl* is the average document length in the text collection from which documents are drawn. $k_1$ and $b$ are free parameters, usually chosen as $k_1 = 2.0$ and $b = 0.75$. IDF($q_i$) is the IDF weight of the query term $q_i$.

The scores generated by retrieval system are probably not in (0, 1), thus normalization is necessary. The following formula was used in our runs:

$$s=\frac{score-\lfloor \min \rfloor}{\max-\min+1} \quad (4)$$

Here, score is the original score, s is the final score. Max and min are the maximum and minimum of the original score respectively.

## 5. Allocating Homepages for Entities

This stage was the last part of our task. Before this stage, four runs which came from the retrieving methods were produced. In each run, there were a number of entity results for 70 topics. Those results in each topic were ranked by the score, like the follows:

```
……
1 Q0 clueweb09-en0012-54-05924 6 0.884194 PRIS1 T_Mobile_PO
1 Q0 clueweb09-en0010-63-31612 7 0.881037 PRIS1 E_Plus
1 Q0 clueweb09-en0012-54-05919 8 0.880843 PRIS1 T_Mobile_US
……
```

Figure 5. A example of a run.

Our four runs adopted the same processing method. It is as follows: First, for any line in a run, parse out the document id and entity name. Second, construct a URL using entity name and get the html by Google. The html contains tens of thousands of relevant results of this entity, but only the first five results are needed. Title and URL are extracted. Some character strings can be found in title and URL, which can be used to measure the possibility of a result being the homepage of the entity. We give a score for the first five results, then resort them in descending order according to the scores they get. Third, search docid from URL database. The homepage of the entity is allocated when a docid is found. The default homepage document id in the original entity run is

replaced with the new docid. Finally, keep the first docid-duplicated entity, and remove all the others if duplicates of document ids in a topic are found.

## 6. Experiments and Submitted Runs

Our experimental data, the English subset of the "Category A" of ClueWeb09, contain about 500 million English pages. For each query, we returned up to 100 related entities. Supporting documents were omitted. The four runs and descriptions are shown in table 1.

Table 1. Runs and descriptions.

| Run Tag | Description |
|---|---|
| PRIS1 | Document-Centered Model by Indri |
| PRIS2 | Document-Centered Model by Indri,<br>giving priority to entities extracted manually |
| PRIS3 | Entity-Centered Model by Indri |
| PRIS4 | Entity-Centered Model by Lemur (BM25) |

Because the evaluation results of other groups have not been published, we only compare our runs to each other. In total 50 topics of 2010 task, 47 ones are evaluated. Here we list the results of submitted runs in table 2. The PRIS2 run presents the best performance in both nDCG_R and P@10.

Table 2. nDCG_R and P@10 of submitted runs.

| | PRIS1 | PRIS2 | PRIS3 | PRIS4 |
|---|---|---|---|---|
| nDCG_R | 0.2158 | **0.2846** | 0.216 | 0.1761 |
| P@10 | 0.1745 | **0.2489** | 0.1766 | 0.1426 |

According to the baseline (best, median and worst), we mapped our result (PRIS2) into the different intervals. Table 3 shows the distributions of nDCG_R. In 2010, 36 topics were above the average level. More than 91% of all topics achieved or surpassed median values. While in 2009, over half of them were below it. Table 4 proves that the improvement of finding primary homepages. Most are in the [best, median] interval. Last year the topics equaling median took up 80%, but most of them were not allocated a correct homepage. 312 out of 779 primary homepages were found, which is taken up over 40%.

Table 3. The nDCG_R distribution of PRIS2 in the last two years.

| | Best | Best~Median | Median | Others |
|---|---|---|---|---|
| 2010 | 1 | **35** | 7 | 4 |
| | 2.13% | **74.47%** | 14.89% | 8.51% |
| 2009 | 0.00% | 35.00% | 10.00% | **55.00%** |

Table 4. The P@10 distribution of PRIS2 in the last two years.

| | Best | Best~Median | Median | Others |
|---|---|---|---|---|
| 2010 | 2 | **29** | 15 | 1 |
| | 4.26% | **61.70%** | 31.91% | 2.13% |
| 2009 | 5.00% | 10.00% | **80.00%** | 5.00% |

The improvement of this year is mainly due to some key details. The content of topics and collection were mined more deeply. Appropriate rules tools, and entity lists were constructed for entity extraction. More flexible retrieval models and ways of ranking entities were tried in the experiments. Several resources and methods were used for allocating primary homepages. In contrast to the improvement in the finding of primary homepages, relevant homepages were less found. This exposed the weak point of our work, putting focus only on the primary ones.

## 7. Conclusions and Future Work

It's the second year we participating the REF task. By integrating some new methods we improved our system. By CMU search engine, we eased the job of getting information from the huge data set ClueWeb. At the stage of entity extraction, special web page, NER tool and entity list generated by some rules were properly taken into account. ECM was also introduced for model improvement. Finally, we used Google and URL database to refine homepages of ranked entities. In the future, we'll pay more attention to mining the context of entities and finding relevant homepages.

## References

[1] TREC Entity 2010 guidelines. 2010.
[2] Rianne Kaptein, Marijn Koolen. Result Diversity and Entity Ranking Experiments: Anchors, Links, Text and Wikipedia. In proceedings of The Eighteenth Text REtrieval Conference (TREC 2009)
[3] Zhanyi Wang, Dongxin Liu. BUPT at TREC 2009: Entity Track. In proceedings of The Eighteenth Text REtrieval Conference (TREC 2009)