ICTNET at Entity ELC TREC 2011

Bingyang Liu[1, 2], Fan Yang[1, 2], Lei Cao[1, 2], Xueqi Cheng[1], Yue Liu[1], Hongbo Xu[1]
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190
2. Graduate School of Chinese Academy of Sciences, Beijing, 100190

1. Introduction

The overall aim of the Entity track is to evaluate entity-related searches on Web data. Entity List Completion (ELC) addresses essentially the same task as Related Entity Finding (REF) does: finding entities that are engaged in a specific relation with an input entity. There are two main differences to REF:

Entities are not represented by their homepages, but by a unique URI (from a specific collection, a sample from the Linked Open Data cloud).

A number of entity homepages (i.e., ClueWeb docIDs) are made available as part of the topic definition, as examples of known relevant answers.The ELC task then is defined as follows:

Given an information need and a list of known relevant entity homepages, return a list of relevant entity URIs from a specific collection of Linked Open Data.

2. Data Format

The Sindice-2011 datasets provides information in the form of RDF triples about entities, i.e., the incoming and the outgoing relations centered around the entity identifier. The dataset is available in two different formats: structured around documents (Sindice-DE) and structured around entities (Sindice-ED). These two sub-collections are built from the same crawl; however, blank nodes are filtered out in Sindice-ED, therefore it is a subset of Sindice-DE.

3. Data Preparation

In this task, we use Lucene to index and retrieve initial results from Sindice 2011, as the entity tool provided by Renaud Delbru cannot run in our local environment. But we still thank him a lot for his dedicated work.

For each query in the 50 topics, we searched the entity_name and every example entity using Lucene. Take the second query (22) as an example:

```
<query>
<num>22</num>
<entity_name>Organization of Petroleum Exporting Countries (OPEC)</entity_name>
<examples>
  <entity>
    <name>qatar</name>
  </entity>
  <entity>
    <name>iran</name>
  </entity>
</examples>
</query>
```

The query above does not contain every tag in the topics file, as the space is limited in this report. We firstly used the underlined entities and searched them in Lucene to get the top 100 results. After this step, we got 644 separated result files.

4. Build the Directed Graph

For each topic, we name the entity in tag <entity_name> as source entity (SE), and name the entity in tag <examples><entity><name> as example entity (EE). From the search result of SE, we get two set of edges. Edges in the first set belong to the outgoing triples, which are from the SE to other entities. Edges in the second set belong to the incoming triples, which are from other entities to the SE. We can get the two sets of edges of each EE in the same way. It is assumed that these edges which are in the results of the SE and every EE will share some common vertexes. Then these relative entities are connected by a directed graph.

5. Expand the Result Set

As we get a directed graph above, we can use the graph to find more related result entities (RE). RE equals EE in the beginning.We color the SE in red and color all the RE in blue in the graph. Then the edges from red to blue are colored in green. These green edges are 1^{st}-step predicates set. Then we add some similar predicates to the set, like if 'locate' is in the predicates set, 'location' will be added. This is done by manual work. The completed predicates' set is called outgoing-set. Another set called incoming-set which contains the passive of the outgoing-set, is build by manual work, too. For example, if 'establish' is in the outgoing-set, then we will add 'established', 'creator' to the incoming-set. Now we can add all the entities which are connected from SE by any edge in the outgoing-set. As the same, entities which are connected to SE by any edge in the incoming-set are also added to the RE.

The operations above can be run several times to expand RE. But in the experiments we carried out, more than one run will bring in too much errors to make the precision very low.

6. Ranking

As the edges are retrieved using Lucene, we assign a value to each edge according to its score in Lucene. The initial EE's are assigned 1.0. Each edges added by manual work are assigned with its related edge's score times 0.8. As all the useful edges and vertexes are assigned with scores, the ranking score of any new entity is basically the product of every edge and vertex on its discovery path.

7. Conclusion

The results are easily affected by manual work which leads the most important part in the method we applied. This is very unstable and not satisfying. Although we did submit the final results, we consider it as a failure.

8. Acknowledgements

We thank all the organizers of TREC Entity Track and NIST. We appreciate the efforts of Krisztian Balog, Pavel Serdyukov, Arjen P. de Vries and specially thank Renaud Delbru for his tool. Thanks for the financial support from National Natural Science Foundation of China (No. 60903139, No.60873243) and 863 projects 2010AA012502, 2010AA012503.

9. References:

[1] http://trec.nist.gov/tracks.html

[2] http://ilps.science.uva.nl/trec-entity

[3] http://data.sindice.com/trec2011/documentation.html

[4] https://github.com/rdelbru/trec-entity-tool

[5] Bhavana Dalvi, Jamie Callan, William Cohen, Entity List Completion Using Set Expansion Techniques, Feb 28, 2011