Fasilkom UI at TREC 2011
Entity List Completion Task

Ananda Budi Prasetya, Hapnes Toba, Mirna Adriani, Hisar Maruli Manurung

Faculty of Computer Science
University of Indonesia
Depok Campus, Depok 16424, Indonesia
{ananda.budi,hapnes.toba}@ui.ac.id,{mirna,maruli}@cs.ui.ac.id

ABSTRACT
In this paper we describe our submissions to the TREC 2011 Entity Track. We have experimented with several combined approaches to search the entity candidates, i.e.: by resolving the linguistic relation of the given entity, query expansion by example to broader the retrieval results, and ontology approach to identify the named entity from the search result snippets and to retrieved the candidate entity. We rank the entity candidates based on the frequency of each entity in the web search result snippets. At the end of our system architecture we performed phrase-based search mechanism in the Sindice dump collection to retrieve specific URIs for the final entity list.

1. INTRODUCTION
One of the tracks joined by Universitas Indonesia in the 2011 TREC conference is the Entity List Completion (ELC) Track. The objective of the task is to get the list of relevant entities from a given information need (i.e.: the query narrative in natural language description) and a list of known relevant entity homepages (i.e.: the example entities), and return the list of relevant URI from each relevant entities. The challenge of this task is how to return the list of relevant entities and URI's from specific document collections, respectively the ClueWeb09 and Linked Open Data (Sindice Dump) collection[1]. Table 1 gives an example query from ELC 2011.

A number of successful approaches from last year results have inspired our approach for this year ELC challenge [1]. Dalvi, et al. [2] were using a two-stage retrieval approach to retrieve candidate entities. In the first step, they utilized the 'target entity' as a query to retrieve web documents, and then by using regular expression they retrieved the candidates from the text of the web documents. The next step, they ranked the entity based on similarity of the candidate entities and the target entity. Fang et al [3] were using unified probabilistic framework to retrieve candidate entities, and utilized specific information in the query narrative. They also used Billion Triple Challenge (BTC) dataset to retrieve the entity by using Lemur as the retrieval engine.

In this paper, we propose an ontology-based named entity recognizer mechanism to retrieve related entities. Our approach use an unsupervised learning named entity recognizer, i.e.: the DBPedia ontology[2], to identify entities from a plain text. DBPedia ontology is an ontology populated with concepts and categories from Wikipedia.

In this paper we reported our system architecture during the experiments, and the special treatments in each submitted runs.

Table 1 ELC 2011 Query

<query>
<num>21</num>
<entity_name>Bethesda, Maryland</entity_name>
<entity_homepage id="clueweb09-…">http://www.bethesda.org/</entity_homepage>
<target_entity>location</target_entity>
<target_type_dbpedia>Building</target_type_dbpedia>
<narrative>What art galleries are located in Bethesda, Maryland?</narrative>
<examples>
<entity>
<homepage id="clueweb09-…">
http://www.discoverygalleries.com/
</homepage>
<name>discovery galleries</name>
</entity>
</examples>

2. SYSTEM ARCHITECTURE
Our system consists of the following main components (Figure 1): query processing, entity recognition and retrieval, and URI identification.

Figure 1 General Scenario

Explanation of each component will be given in the following sub-sections.

2.1 Query Processing
In the query-processing component, each ELC query is parsed to determine the entity name, target entity, DBPedia target type, narrative, and the entity examples.

The objective of this step is to identify the context description, i.e.: the nouns (NN, NNP, NNPS, NNS) and cardinal numbers (CD), that we considered as the information need.

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

[2] http://dbpedia.org

The narrative of the query is further processed in order to resolve the linguistic relation of the given entity by using a part-of-speech (POS) tagger, we use Stanford POS Tagger[3] during the experiments, see Figure 2.

Each term in the narrative, which related to a specific context description and the given entity examples will be used as the query terms in the ClueWeb09 web service, as a kind of query expansion . For example, in query #21*(What art galleries are located in Bethesda, Maryland?)*, the query terms which passed into the ClueWeb09 web service will be:

QUERY #21 + Expansion by Example Entities
art gallery bethesda maryland
+ discovery galleries
+ glen echo park
+ the fraser gallery
+ washington school of photography
+ waverly street gallery
+ creative partners gallery
+ yellow barn studio and gallery
+ orchard gallery
+ hendricks art collection limited
+ marin-price galleries

2.2 Entity Recognition and Retrieval

In the entity retrieval component, we delivered the top-100 snippets of the ClueWeb09 results into the DBPedia Spotlight[4] web service, which is based on the DBPedia Ontology. Our main objective is to identify the desired entity target type as required by the ELC query. After this entity identification step, we count the frequency of each entity, which occurs in the ClueWeb09 snippet results and normalized it by a factor of 100.

We assume that frequency indicates the level of similarity between an entity candidate, the examples and the query description. We ranked the frequencies in decreasing order to form a list of entity candidates, see Figure 3. In this manner we expected to retrieve some new entities, which simultaneously mixed with the related given entities in the query example.

2.3 URI's Identification

At the final stage, we perform search in the link open data (LOD) collection, i.e. the Sindice dump for each entity candidate. During this search, we used the entity-document (ED) centric approach because we were interested in finding entity across multiple contexts [4, 5].

Our specific strategy is to perform in-depth retrieval by using 'OR-like' function for each entity. The scenario and result from URI's identification process can be seen in Figure 4. We considered each term occurs in an entity as independence terms during the search, for instance the entity *'The Gallery at Market East'*, will be queried as *'gallery OR market OR east'* during the search. The main objective of this strategy is to retrieve all possible relevant documents, which contain part of the entity terms.

To validate the final URI, we perform a phrase checking mechanism. It compares all of the terms occur in an entity to the content of a retrieved URI. If the entity terms were found (exact match) in an URI, then it will be considered as the final answer. For example the entity *'The Gallery at Market East'*, has a validated URI *'http://dbpedia.org/resource/The_Gallery_at_Market_East'*.

3. Submitted Runs

All of our submissions are based on the system description described in the previous section. Our submitted run setup can be seen in Table2.

Table 2. Submitted Run Setup

Characteristic	Run 1	Run 2	Run 3
Example in final list	No	Yes	Yes
Score function	Frequency-based	Frequency-based	Frequency-based and penalization of example entities

In this section we reported specific treatments that we have performed in each run.

3.1 Run 1

In this run, we excluded the example entities during the candidate list development. On the contrary, we included all of the entities that have the same target entity type as mentioned in the ELC query into the list, and rank them in decreasing order based on their normalized term frequency scores.

As an example, the top-10 entity list candidate for query #21 in this run is given in Table 3.

3.2 Run 2

The difference between Run 1 and Run 2 lies in the treatment of the example entities from the original ELC query.

In this run, we included the example entities in the candidate list, and simultaneously rank them with the retrieved candidate entities – based on their term frequency scores – to form the final list.

As an example, the top-10 entity list candidate for query #21 in this run is given in Table 4.

3.3 Run 3

The difference between Run 2 and Run 3 lies in the relevance score calculation.

In this run we re-ranked the entities by penalizing the score of the example entities by a constant factor. We considered that the example entities have lower priority than the retrieved candidate entities. In order to decrease the relevance score of the example entities, we subtracted the original score of a known example entity to a constant number (40 in our case). The chosen constant number must be big enough to increase the rank of new entities in the candidate list.

As an example, the top-10 entity list candidate for query #21 in this run is given in Table 5.

[3] http://nlp.stanford.edu/software/tagger.shtml

[4] http://spotlight.dbpedia.org

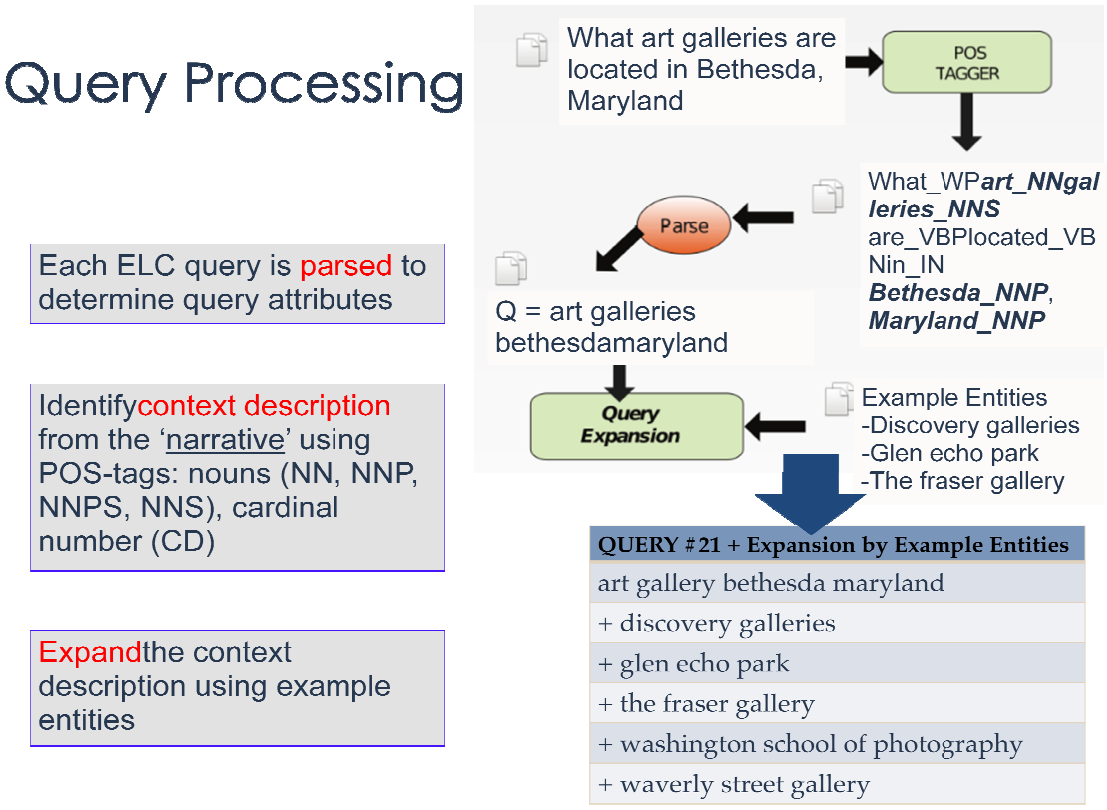

Figure 2. POS Tagger to Identify the Context Description

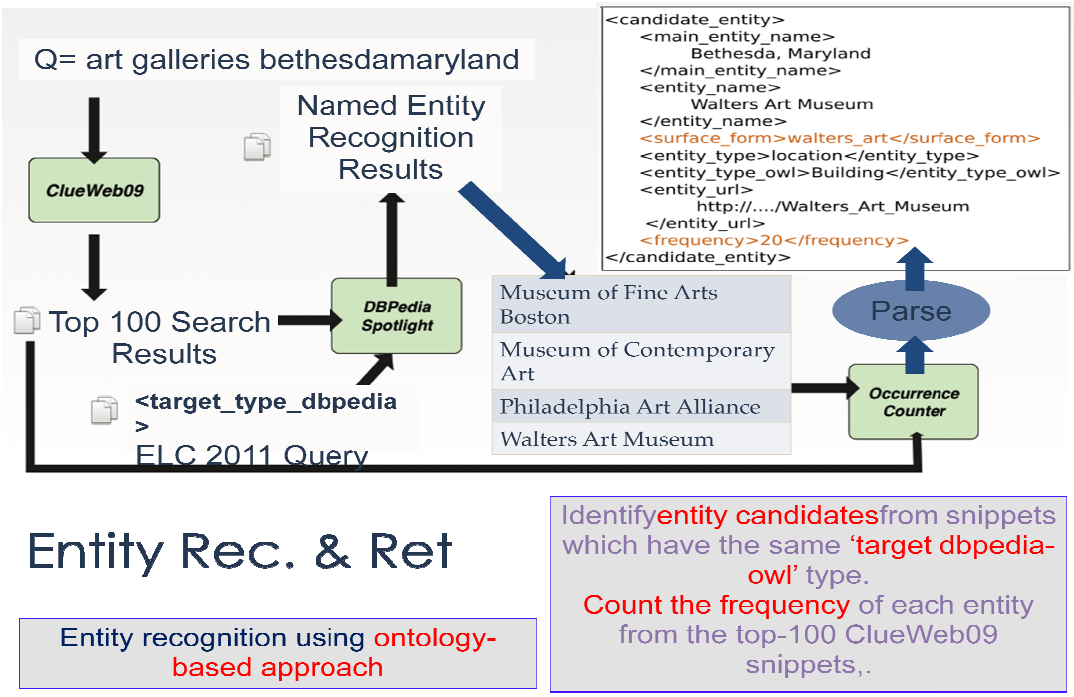

Figure 3. Entity Recognition by Using DBPedia Spotlight

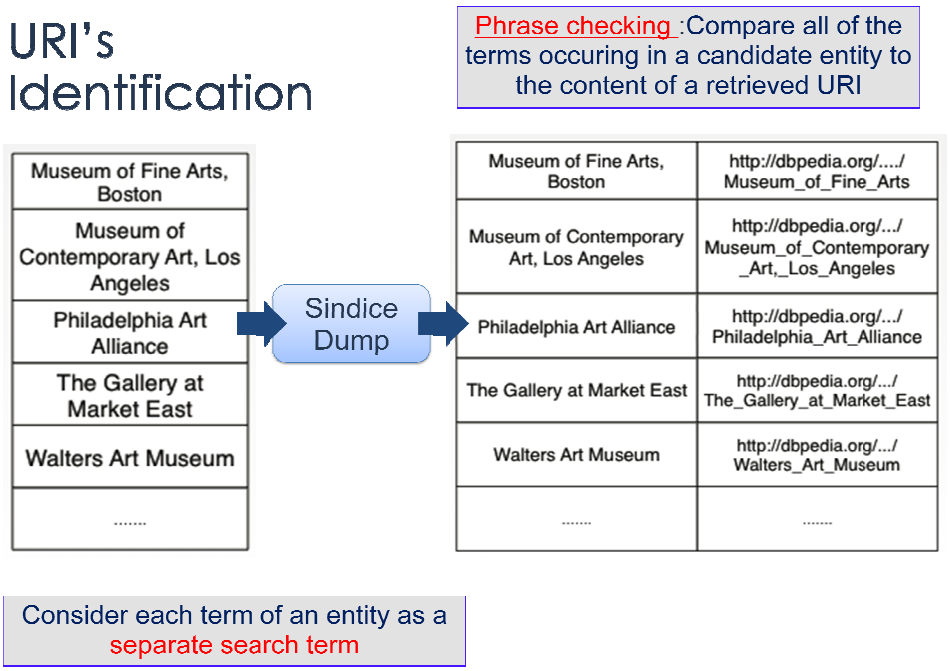

Figure 4. URI's Identification

Table 3. Top-10 Entity List Candidate for Query #21 in Run 1

No.	Entity List Candidate	Normalized Frequency	Entity Status
1.	Museum of Fine Arts, Boston	0.46	New
2.	Museum of Contemporary Art, Los Angeles	0.34	New
3.	Philadelphia Art Alliance	0.34	New
4.	The Gallery at Market East	0.26	New
5.	Walters Art Museum	0.17	New
6.	Dayton International Airport	0.15	New
7.	Delaware Center for the Contemporary Arts	0.14	New
8.	Baltimore Museum of Art	0.09	New
9.	Metropolitan Museum of Art	0.07	New
10.	Brigham Young University Museum of Art	0.05	New

Table 4. Top-10 of Entity List Candidate Query #21 in Run 2

No.	Entity List Candidate	Normalized Frequency	Entity Status
1.	The Fraser Gallery	0.68	Example
2.	Museum of Fine Arts, Boston	0.46	New
3.	Marin-price Galleries	0.41	Example
4.	Glen Echo Park	0.38	Example
5.	Museum of Contemporary Art, Los Angeles	0.34	New
6.	Philadelphia Art Alliance	0.34	New
7.	Creative Partners Gallery	0.33	Example
8.	The Gallery at Market East	0.26	New
9.	Washington School of Photography	0.18	Example
10.	Walters Art Museum	0.17	New

Table 5. Top-10 Entity List Candidate for Query #21 in Run 3

No.	Entity List Candidate	Normalized Frequency	Entity Status
1.	Museum of Fine Arts, Boston	0.46	New
2.	Museum of Contemporary Art, Los Angeles	0.34	New
3.	Philadelphia Art Alliance	0.34	New
4.	The Fraser Gallery	0.28	Example
5.	The Gallery at Market East	0.26	New
6.	Walters Art Museum	0.17	New
7.	Dayton International Airport	0.15	New
8.	Delaware Center for the Contemporary Arts	0.14	New
9.	Baltimore Museum of Art	0.09	New
10.	Metropolitan Museum of Art	0.07	New

4. CONCLUSIONS

In this paper we have outlined our approach in the TREC 2011 Entity track. We have demonstrated that frequency-based entity scoring, combined with a lightweight linguistic and ontology processing, can be used to finding new entities to complete a given related entities.

We have experienced some dilemmas by using the ClueWeb09 web service. In one hand, we have no direct control to the indexing and retrieval strategy, but in the other hand, we are challenged to deal with a very huge data collection, around 25 TB of uncompressed data.

Due to the lack of evaluation judgments, we have not analyzed the rank relevance measure of our approach yet.

5. REFERENCES

[1] Balog, Krisztian, Serdyukov, Pavel , and de Vries, Arjen P. “Overview of the TREC 2010 Entity Track”, 2010.

[2] Dalvi, Bhavana, Callan, Jamie, and Cohen, William. “Entity List Completion Using Set Expansion Techniques”, Proceedings of the Nineteenth Text Retrieval Conference 2010.

[3] Fang, Yi, Si, Luo, Somasundaram, Naveen, Yu, Zhengtao, and Xian, Yan-tuan. “Probabilistic Framework for Matching Types between Candidate and Target Entities”, Proceedings of the Nineteenth Text Retrieval Conference 2010.

[4] Campinas, S., Ceccarelli D., Perry, T.E.,Delbru, R., Balog, K., Tummarello, G., “The Sindice-2011 Dataset for Entity-Oriented Search in the Web of Data”, EOS, SIGIR Workshop, Beijing, China, 2011.

[5] Delbru, Renaud, Toupikov, Nickolai, Catasta, Michelle, and Tummarello, Giovannie. “A Node Indexing Scheme for Web Entity Retrieval”, European FP7 project Okkam - Enabling a Web of Entities (contract no. ICT-215032), 2009.