A journey in Entity Related Retrieval for TREC 2009

Jagadish Pamarthi , Guangxu Zhou, and Coskun Bayrak

Department of Computer Science
University of Arkansas at Little Rock
Little Rock, AR

Abstract

The focus of this paper is to present the results obtained as a result of performing entity information retrieval, namely the home pages of products, organizations and persons. The preliminary results, based on the Indri Search Engine, of this study and experimentation were presented at the Entity Track in TREC 2009. Indri Search Engine is an efficient and effective open source tool, which is operated by indri query language in any windows or UNIX based platform. Indri is based on the inference network framework and supports structured queries.

Introduction

The Entity Track, which is motivated from the Enterprise Track, was introduced for finding the home pages of entities like products, organizations or persons. The Enterprise track has provided a platform to look at one specific entity from two directions. The first one is expert finding, which finds entities in the collection (retrieving entities in particular context). The second is expert profiling, which gets insights about entities (retrieving the context for a given entity). Historically, the entity is something which has a home page. Therefore, Persons, Products, and Organizations were the tree types of entities to be considered in the process of information retrieval. During the study each participant is required to submit results for the given queries of persons, products, and organizations. For Entity track, ClueWeb09 dataset composed of 1 billion pages in 10 languages was used. Based on the instructions provided, “category B” subset, which contains about 50 million English pages for the entity track, was used. For indexing, the Lemur tool kit in Red Hat Enterprise Linux platform and for retrieval Indri query language were deployed for the web pages. The files were indexed to form two repositories. The first one contains the Wikipedia pages and the second consists of pages other than Wikipedia. Wikipedia pages were given as optional therefore carried less importance. Since it took nearly two days, the indexing of all the WARC files in Indri search engine was timely intensive process. Indri is highly efficient and effective language for retrieving the pages indexed and it supports structured queries. It gives more efficient results, when we indexed more pages into the search engine, when compared to the less number of indexed pages. Using different approaches/procedures, up to four different results/runs (UALRCB09r1, UALRCB09r2, UALRCB09r3 and UALRCB09r4 from the lowest priority to the highest priority)

Entity Relation Finding Task

Entity related finding considers the fact that given the name and homepage of an entity, as well as the type of the target entity, find related entities that are of target type. The input contains the name and homepage of an entity, type of target entity, and context for the search. The output documents must contain homepages of the target entities and the supporting documents. The example format of the input has the following form.

<query>
<entity_name>kimi raikkonen</entity_name>
<entity_URL>http://www.kimiraikkonen.com/</entity_URL>
<narrative>I'd like to know which organizations are sponsoring kimi</narriative>
</query>

Now the solution for this query is extracted by using the Indri search engine. The source of the index files for the repositories, which contain the Wikipedia pages and the other repository not containing the Wikipedia pages were given. In order to obtain good scores and results for the Information Retrieval different combinations were experimented. The combinations include the terms which can extract the home page of the given entity and rejects the terms (like Wikipedia pages), which doesn't retrieve the home page. Figure 1 represents the systematic approach of retrieving the home page of a given entity. The search interface will be at the user end and it will be accessed by giving the related entities. The query will be processed into the indri search engine and then the database will be accessed and the required home page of the entity along with the scores will be retrieved.

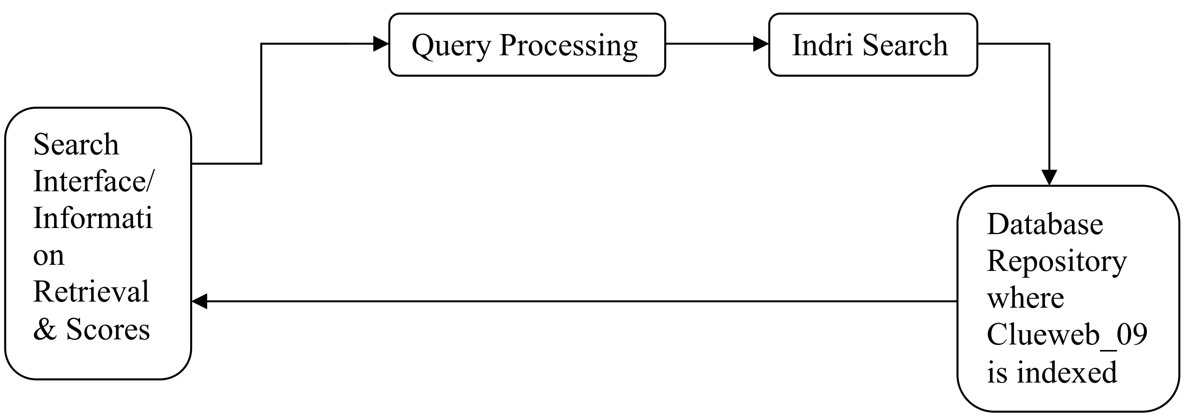

Figure 1: Systematic approach for the entity related finding

Base Run with Simple Query

For indexing the Clueweb09 the Lemur Tool Kit version 10 was installed, since the ClueWeb09 collection was in the WARC format. The latest version was designed for the indexing the WARC files directly. UALRCB09r1 is our base run, which was build with simple queries for all the topics run on the Clueweb09 collection. For example, topic 1 is given to find "carriers that blackberry make phones for". It is converted to the simple query and can be written as *#combine (carriers blackberry makes phones)*. Then the results along with the scores will be displayed on the search interface of each and every retrieved page. The "carriers of the blackberry" task was initially performed. Then we gave the queries to find the home pages of the carriers of blackberry using different queries using the different keywords containing in the required home pages. The graph of the nDCG_R for each and every topic obtained is shown in the Figure 2.

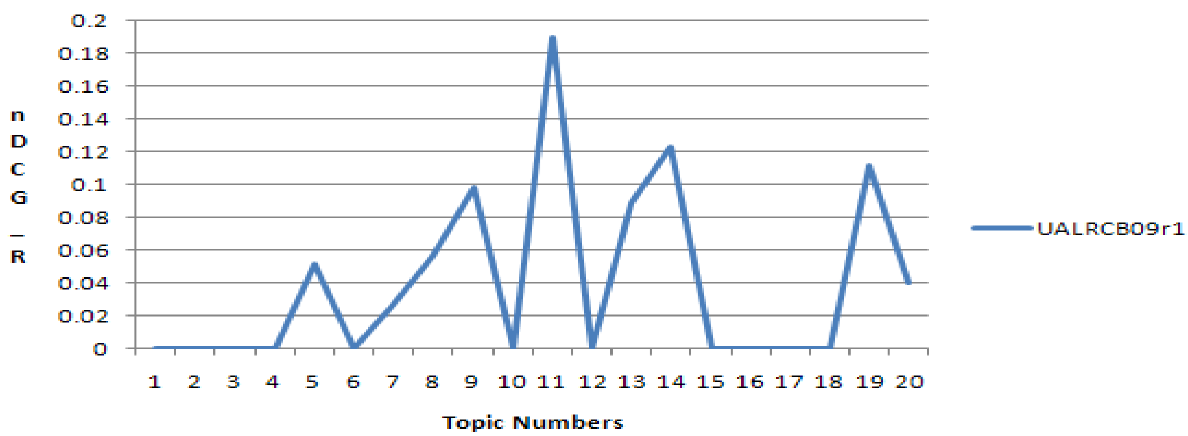

Figure 2: Graph of nDCG_R for UALRCB09r1

Run with Complex Queries

Now the tasks were done by using complex queries for all the remaining three runs. With these three runs the pages from the ClueWeb09 collection, which was indexed using the Lemur were also retrieved. In addition, the queries were defined according to the output required. For instance, first all the pages including the Wikipedia pages were retrieved and then the compiled list was filtered to eliminate the Wikipedia pages via query (since the Wikipedia pages were optional). At least 10 pages for every query were retrieved: the first two were taken as the primary pages and the next pages as the supporting documents. To give an idea of the complex structured queries, let's consider the following example.

#weight (0.8 #filrej(Wikipedia #combine(carriers blackberry makes phones) 0.1 #combine(#1(carriers blackberry) #1(blackberry phones) home page) 0.1 #combine(#uw8(blackberry carriers home page))

All the tasks are performed once again using the complex queries and the outcome is called UALRCB09r2. The content of UALRCB09r2 contains less efficient results when compared to the UALRCB09r1. The graph of the UALRCB09r2 with nDCG_R for all the topics was represented in Figure 3.

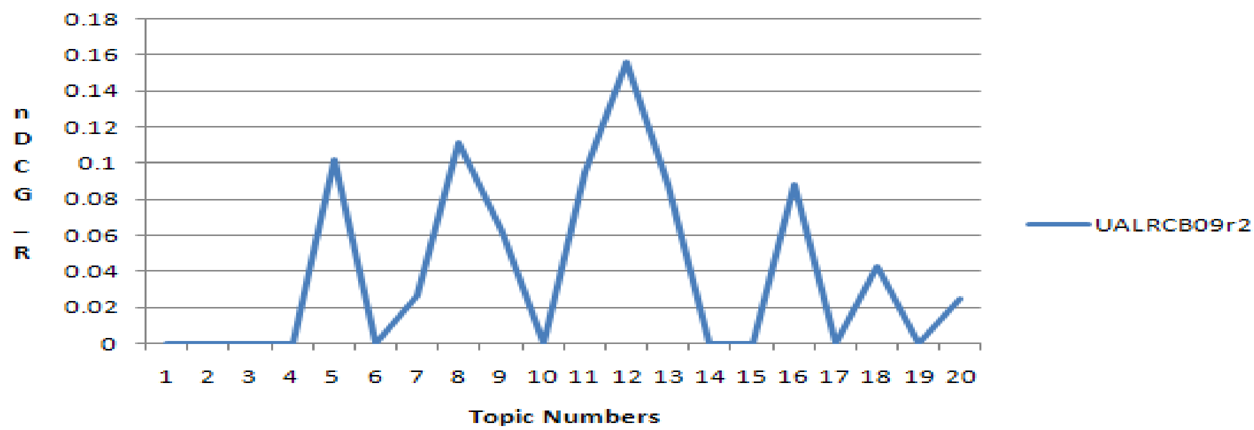

Figure 3: Graph of nDCG_R for UALRCB09r2

UALRCB09r2 was performed with the complex queries and now a new approach with different query structure is being considered for the UALRCB09r3. After adjusting the precision, more efficient results were obtained, when compared to the UALRCB09r2. The results were depicted as a graph in Figure 4. Finally the last run, called UALRCB09r4, were performed as a result of a small refinement to see if the outcome can be further improved. Figure 4 contains the graph of UALRCB09r3 and nDCG_R for all of the topics given.

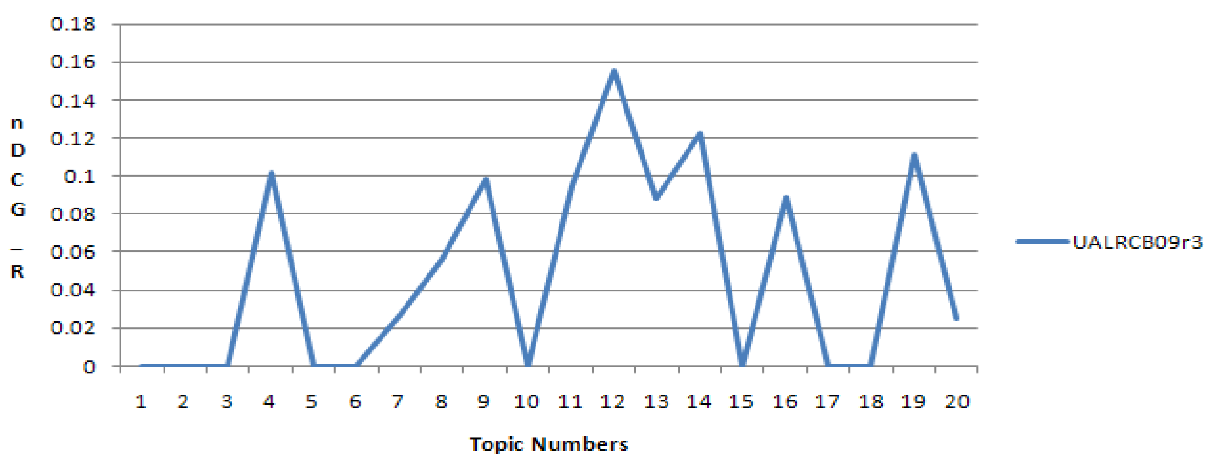

Figure 4: Graph of nDCG_R for UALRCB09r3

Results

The results obtained for the entity track were tabulated in the Table 1 for all of the four runs, namely UALRCB09r1 through UALRCB09r4. The experiments conducted under this investigation shed light not only for the task undertaken this year but also set a reliable foundation for the upcoming studies in this area. The graph of nDCG_R for all the runs was shown in the Figure 5.

Table 1: Results for all the twenty topics

Topic	# Pri	NDCG at R Best	NDCG at R Median	NDCG at R Worst	Primary Retr @ 10 Best	Primary Retr @ 10 Median	Primary Retr @ 10 Worst
1	10	0.2992	0.0597	0.0000	2	0	0
2	1	0.4262	0.1012	0.0000	1	0	0
3	1	0.6388	0.0000	0.0000	1	0	0
4	5	0.2982	0.0417	0.0000	3	0	0
5	8	0.3697	0.1119	0.0000	4	1	0
6	4	0.2844	0.1168	0.0000	2	0	0
7	33	0.2955	0.0661	0.0000	8	0	0
8	13	0.4838	0.0559	0.0000	5	0	0
9	2	0.3728	0.1602	0.0000	2	0	0
10	15	0.4596	0.0598	0.0000	8	0	0
11	8	0.3668	0.0499	0.0000	3	0	0
12	13	0.3663	0.0469	0.0000	3	0	0
13	4	0.2815	0.0884	0.0000	1	0	0
14	4	0.6842	0.0772	0.0000	4	0	0
15	9	0.5796	0.0714	0.0000	6	0	0
16	9	0.4319	0.0000	0.0000	6	0	0
17	18	0.3379	0.0816	0.0000	5	0	0
18	3	0.4312	0.1414	0.0000	1	0	0
19	3	0.3647	0.0000	0.0000	2	0	0
20	4	0.4243	0.1725	0.0000	3	0	0

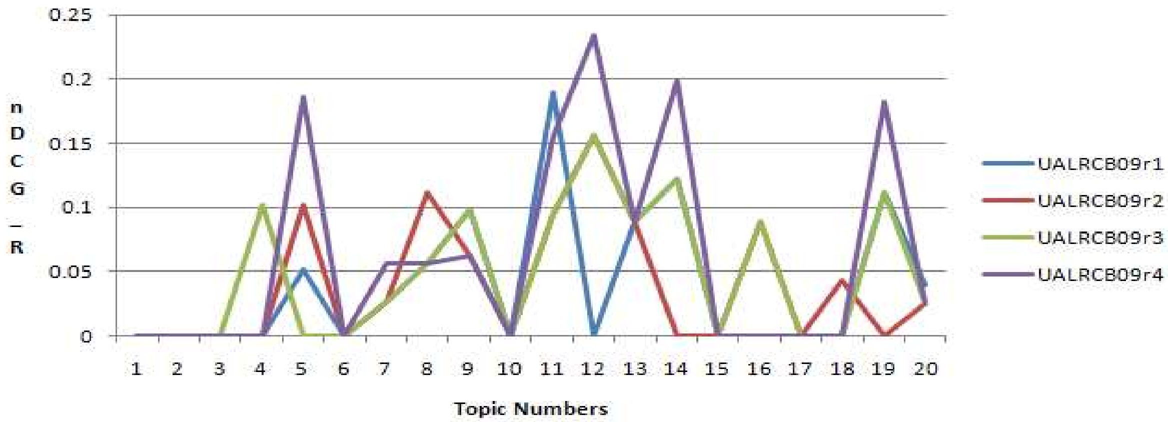

Figure 5: Graph of nDCG_R for all the runs

Conclusion

In the entity retrieval study we investigated how the Indri search engine performs for the different queries to retrieve the required entities in noisy web environments. As a result four officials run showing the success ratio in getting the good and efficient output for retrieving the home pages were analyzed. Positive results were obtained by using the complex queries. When doing the information retrieval on the ClueWeb09 we used porter stemming on unstopped data. The Indri search engine is both efficient and effective on such large scale collections. Indri indexed the 50 million documents; 5TB ClueWeb09 collection uncompressed in 2days and processed approximately one query every second. In terms of effectiveness, phrase expansion via Indri's structured query operators proved to be a powerful asset. Despite all of this, we hope to improve our system for next year. There are a number of things we aim to explore, including faster indexing, improved query processing times, looking into further use of complex queries, more effective query expansion techniques for noisy data.

References

1. http://ilps.science.uva.nl/trec-entity/
2. http://www.lemurproject.org/
3. D. Metzler and W.B. Croft. 2005. A markov random field model for term dependencies. In *Proceedings of SIGIR 2005*, pages 472–479.
4. D. Metzler, T. Strohman, Y. Zhou, and W.B. Croft. 2006. Indri at TREC 2005: Terabyte track. In *Proceedings of 2005 Text REtrieval Conference (TREC 2005)*.
5. T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. 2005. Indri: A language model-based serach engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*.