Delft University at the TREC 2009 Entity Track: Ranking Wikipedia Entities

Pavel Serdyukov, Arjen de Vries
Delft University of Technology
Delft, The Netherlands
p.serdyukov@tudelft.nl, arjen@acm.org

ABSTRACT

This paper describes the details of our participation in Entity track of the TREC 2009.

1. INTRODUCTION

Entity ranking is a novel TREC task introduced this year and posing challenges similar to those already well-known in web retrieval research community. We present a system which leverages a number of popular web retrieval techniques, utilizes existing knowledge bases and relies on various task-specific heuristics to produce high quality runs. Since we had no training queries with relevant web-pages contained in Category B part of ClueWeb09 collection, we focused on using various strategies rather than on using one kind of approach with different parameter settings. However, in three of four submitted runs we treated Wikipedia part of the collection as the main source of evidence about relevance of entities that can be found on the Web.

2. COLLECTION PROCESSING

2.1 Indexing and querying ClueWeb09

Relying on the most convenient strategy, we indexed WARC bundles containing ClueWeb09 collection using Lemur Toolkit 4.10. We used no stemming and stop-word removal on the collection level. However, we applied a very simple stemming strategy for queries, by adding singulars to the query if there were any plurals. In addition, we used a stopword list with 320 words to exclude stopwords from queries. We used both the title and the narrative to build a query. Finally, we ranked the pages using Language model based approach implemented in Lemur with Dirichlet smoothing ($\mu = 1500$).

In order to have some flexibility in development and also for the sake of better performance, we built several indexes of different purpose:

- Index containing all pages from ClueWeb09 (Category B) collection, except those contained in the bundles named as "enwpXX",
- Index containing all pages from the bundles named as "enwpXX" (Wikipedia pages),
- Index containing anchor text of all pages from the non-Wikipedia part of ClueWeb09 (Category B),
- Index containing anchor text of all pages from the Wikipedia part.

All pages and extracted anchor text were stored in the indexes as well, in order to fetch them dynamically at query time for the further processing.

2.2 Treating Wikipedia as an entity repository

Wikipedia is an online encyclopedia that recently became one of the largest repositories of encyclopedic knowledge, with millions of articles available for a number of languages. English Wikipedia is also a part of the ClueWeb09 (Category B) and represents a spam-free collection of web-pages with dedicated descriptions of around 2,700,000 entities and concepts. While it is not possible to find any entity a user may ask for among those described in Wikipedia, there is still a very high chance that most potentially popular queries can be answered by ranking entities described in this repository.

Wikipedia part of the ClueWeb09 is a collection of raw HTML pages, so, it needs some basic cleaning in order to serve as an entity repository. While we indexed the entire Wikipedia, we always ignored non-article pages (lists, disambiguation, category pages, etc.) at the query post-processing stage. However, we still had a problem of duplicates, since typically a number of URLs, all existing name variants of the same entity(e.g. "`/wiki/Crackberry`" and "`/wiki/Blackberry_Storm`") and all different ClueWeb09 documents, are redirected to one page in Wikipedia (e.g. "`/wiki/Blackberry`"). All these URLs have the same title still. So, since we had to submit only one Wikipedia document per entity, we always selected the URL with the highest number of inlinks within Wikipedia collection among documents with the same title.

3. BASELINE: ANCHOR-BASED ENTITY RANKING

In order to first approach the problem of ranking with the simplest strategy possible, we decided to find entity mentions at the given primary page not using any given dictionary of entities (e.g. the titles of Wikipedia articles). While, the traditional approach would be to run a named entity tagger against the primary web-page of the query entity given, we did not follow this strategy by several reasons. First, such tagging would give us potentially a very long list of entities with still the need to disambiguate many of them. And, second, we would still need to search for primary pages for each of discovered entities and rank them by their relevance. While we still consider worthwhile to follow that path, we decided to rely on the clues left for us by the author of the primary web-page. We considered anchor text of its outlinks as entity names, and the pages linked as primary for these entities. As a result, it left us with no need to disambiguate names and search for primary pages, as well as we had the reason to hope that only important entities with strong relation to the main subject of the page were linked by its author. In order to extract actual entity names from anchors we also removed the following words: "homepage of", "about", "information about". We then found the pages in the collection, ranked them and also relied on the following ranking rules, mainly following the instructions specified in the guidelines for the task:

- For the entities of the type "product" we always ranked pages hosted at the domain of the given primary page higher than external pages,
- For the entities of the type "person" or "organization" we always ranked pages from outside of the domain of the given primary page higher than pages from the same domain.

Since, the primary page itself not always contains actual content, we ranked also the pages at the second level on the link path starting from the primary page.

4. WIKIPEDIA-BASED ENTITY RANKING

4.1 Building a candidate list

Since we considered Wikipedia as the primary repository of entity descriptions, we always started from ranking Wikipedia articles and regarding first 3,000 entities as our candidates. After the basic filtering steps (see Section 2.1), we faced two challenges: finding additional evidence of entity relevance using the entire ClueWeb09 (Category B) and filtering out entities of undesired types.

Thanks to Wikipedia category structure and enthusiasm of Wikipedia contributors, it was pretty straightforward to find articles describing persons. We believe that most biographical articles belong either to the category "Living people" (around 337,000), or to the categories "XXXX births" and "XXXX deaths" . So, we just searched for these strings in the text of retrieved Wikipedia pages to filter out persons or to consider them as the only entities allowed (actually, using #scoreif directive of the Lemur query language). Since we still could not be sure that all relevant entities are properly categorized, we returned them in the ranking as well, but after the matching ones. All approaches described further benefited from this strategy, including the approach using external ontologies to guess about the other types of entities (see Section 4.3).

4.2 Using Wikipedia to find named entities

There are several ways to deal with Wikipedia corpus in order to find relevant named entities: simply rank Wikipedia pages, rank Wikipedia pages whose names appear on the given primary Web page of the query entity or rank Wikipedia pages whose names can be found at the top ranked Web pages. While there are pros and cons for each approach, we built our runs using only the latter two strategies (see Section 5). Despite that we regarded all Web pages from the top ranked set as equally important to promote entities, we tried to bring in some variety in their nature and also prevent *spam* pages from taking over the entire top. So, in the case when we relied on top ranked pages, we always considered 5 top ranked pages from the domain of the given primary page (including the primary page itself) and also top 20 pages from other domains.

4.3 Finding primary Wikipedia pages for the given entity

While our task was to search for the entities related to the one specified in the query, there was a very high chance that we find also Wikipedia article(s) describing the query entity itself. Considering that such an article would highly likely lead also to the important entities related to the query entity, we tried to find it and rank only its outlinks. However, since the risk of selecting a wrong "primary" wikipedia page for the query entity was quite high with potentially dramatic loss in performance, we decided to use two sources of evidence: title/narrative and the primary homepage of the query entity. We proceeded as follows:

1. Retrieved top 5 Wikipedia articles as a candidate set,
2. Extracted all URLs from the "External links" (or "References", if necessary) section of each article, preserving the ranking order specified by its author,
3. Selected the Wikipedia article from the candidate set, whose editor assigned the highest rank to the primary homepage of the query entity.

In other words, we either selected the top ranked article, or one of the most highly ranked articles which was the most related to the primary homepage of the query entity.

4.4 Using existing ontologies to find entities of the desired type

Another possible source of additional performance is the correct classification of entities into four classes (person, organization, product, other) and selecting only the Wikipedia entities matching the given target class. Since, the sufficient amount of training data is not available, we utilize external resources for the filtering purposes, namely DBPedia[1] and Yago[2].

4.4.1 Using DBPedia

DBPedia is a highly structured representation of Wikipedia, describing not only typed relations between Wikipedia entities, but also containing contextual links to other repositories. It also provides a basic ontology[3], ideally suited for the entity ranking task, since it has almost exactly the same classes as allowed in Entity track, at the top of its hierarchy. Particularly, we considered 4 of them: "Organization", "Person", "Work" (actually, artifacts made by humans) and "Drug" (conceptually, could be a sub-class of "Work"). The union of the latter two we consider to be equivalent to the "product" class specified in our queries. However, only around 1 million entities are currently classified into this ontology. So, we only filtered out those entities for which the class is known and it does not match the one specified in the query.

[1] http://dbpedia.org
[2] http://www.mpi-inf.mpg.de/yago-naga/
[3] http://dbpedia.org/ontology/

4.4.2 *Using infoboxes, Yago and Wordnet*

Since only a limited number of Wikipedia articles are categorized into DBpedia ontology, we tackle the filtering problem by building additional "feature space" for its classes using two sources: properties specified in the infobox of almost every article (around 1,750,000), and Wordnet classes assigned by Yago to almost every second article (around 1,800,000). While learning an appropriate classifier is a promising direction of the future work, we applied a simple rule-based filtering approach based on discovered feature sets. We filter out entities with the features that do not belong to the class specified in the query, so there is no any other entity that belongs to the query class and has the same feature.

5. FINDING PRIMARY HOMEPAGES FOR WIKIPEDIA ENTITIES

After the candidate Wikipedia pages are ranked, we still need to look for the canonical names of the entities they describe and also for up to 3 candidate homepages to return. Thanks to authors of Wikipedia pages, we are almost always able to find primary web-pages among the "External links" section, as we did to discover "primary" Wikipedia pages. However, since URLs under "External links" are not only homepages, but also just relevant pages, we consider only the top 3 of them. However, in cases when some of these URLs belong to the same domain, we consider only the highest ranked of them and hence end up with less than 3 pages. Another problem is that Category B part of ClueWeb09 contains far not all pages linked from Wikipedia and hence we need some backup strategy to find the pages we still miss. So, we used the anchor text index for that purpose and searched it using either names of the candidate Wikipedia entities or the most popular anchor text strings associated with their URLs. We also did not allow more than one page from the same domain, but since this time they were ranked not by a human editor, we considered that not the highest ranked, but the shortest URL has more chances to be an entry to the entity homepage.

6. RESULTS

We submitted the following runs:

- **tudwebtop**: Run produced using the approach described in Section 2.2, where every outlink from the given primary web-page is ranked,
- **tudwtop**: Run produced using the approach described in Section 4.1, where those Wikipedia entities that appear at the top web-pages are ranked first,
- **tudpw**: Run produced using the approach described in Section 4.2, where those Wikipedia entities that appear at the discovered primary Wikipedia page are ranked first,
- **tudpwkntop**: Run produced using the approach described in Section 4.3, where Wikipedia entities found using the **tudpw** run are post-filtered using ontologies and top web pages.

We decided to compare runs using the variant of evaluation when wikipedia pages are not left out, since our approaches are mainly built on Wikipedia. We do not report values for all measures for every topic here, but rather than analyzing averaged values, we are more interested in the number of topics for which the approach was one of the best performing. For example, for the representative P@10 measure, we see that **tudpwkntop** was the best for 10 topics, **tudpw** for 6 topics, **tudwtop** for 6 topics, and **tudwebtop** for 2 topics. Note that while **tudwebtop** has shown zero performance for 15 topics out of 20, it was the only approach that worked for 2 topics (8th and 16th topics). Also, all approaches failed to produce non-zero performance for the 3rd topic. In two cases, one of the approaches has shown the maximum possible performance (1,0 for 17th and 18th topics). Averaged measures produced for three different sets of qrels are demonstrated in the following tables.

Run	nDCG_R	P10	rel_ret	pri_ret
tudwebtop	0.1218	0.0600	103	28
tudwtop	0.1244	0.0650	125	50
tudpw	0.1351	0.0950	108	42
tudpwkntop	0.1334	0.1150	108	41

Table 1: Performance of all submitted runs for NGCG_R, P@10 scores, and the number of relevant and primary entities (rel_ret and pri_ret, respectively) retrieved. Official qrels (Wikipedia pages are not considered) are used.

Run	nDCG_R	P10	rel_ret	pri_ret
tudwebtop	0.1009	0.0600	103	28
tudwtop	0.1672	0.2250	168	144
tudpw	0.1767	0.2400	140	122
tudpwkntop	0.1778	0.2700	140	120

Table 2: Performance of all submitted runs for all measures. Wikipedia pages are considered.

Run	nDCG_R	P10	rel_ret	pri_ret
tudwtop	0.2551	0.2150	43	94
tudpw	0.2836	0.2300	32	80
tudpwkntop	0.2826	0.2600	32	79

Table 3: Performance of all submitted runs for all measures. Only Wikipedia pages are considered.

7. RELATED WORK

Entity ranking with the focus on Wikipedia is a well-known task being run at INEX conference[4] since 2007 [1]. The main difference, besides the smaller corpus, is in the larger number of entity types allowed. Besides, they are usually specified on much less abstract level, since each query is supplemented with the actual Wikipedia category related to relevant entities (but not always directly). The usefulness of Wikipedia for finding representative keywords and named entities on web pages is demonstrated in several publications. First, this idea was introduced in the Wikify system, which not only matched keyphrases to Wikipedia entities, but also selected the most important of them using

[4] `http://www.inex.otago.ac.nz/`

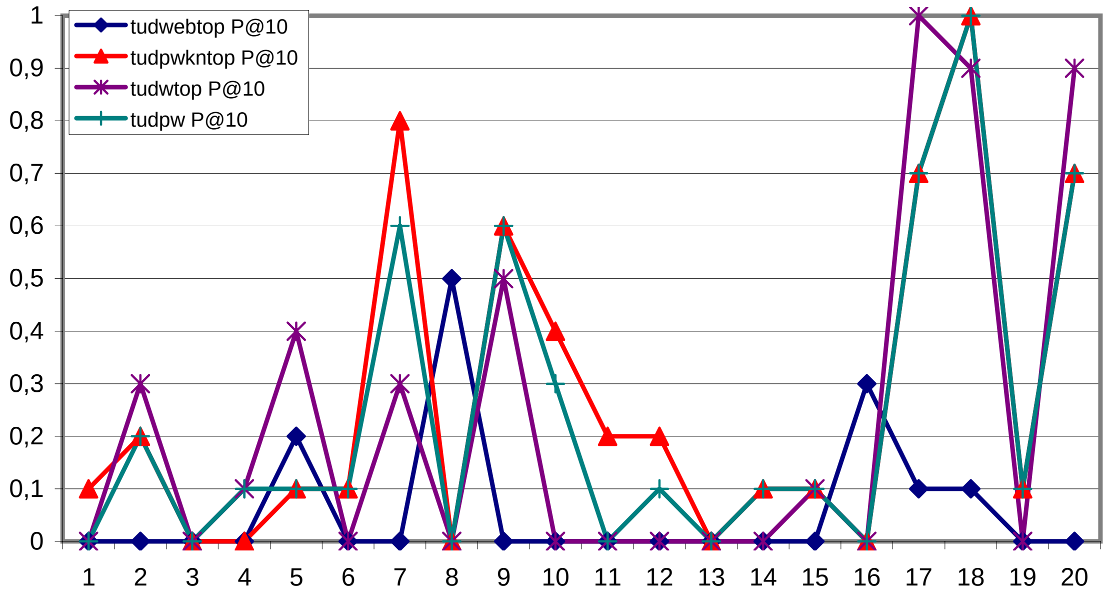

Figure 1: P@10 of all runs per topic for the evaluation considering also Wikipedia pages

the number of article's inlinks [3]. Later, this technique was extended by learning the function of link appropriateness using the number of features: the relatedness of linked entities to the content of the page under study, the minimum depth at which the entity is located in Wikipedias category tree, number of mentions at the page, their locations and proximity to each other [4]. Another recently proposed approach uses the link structure of candidate Wikipedia entities to first group them into clusters and then rank these clusters by the overall importance of their entities, measured using their inlinks, in the way also used by the Wikify system [2]. Despite that some of these methods might be useful also in the present setup, we believe that the task that we deal with suffers much less from the word ambiguity problem, which is the primary issue for the keyword extraction methods. In the case of entity ranking, we have a "query layer" which connects relevant Wikipedia pages with relevant Web pages and hence should implicitly disambiguate entities mentioned at the web-pages. However, it would still be interesting to test the value of link-based entity authority and number of mentions for detection of relevant entities.

8. CONCLUSIONS

We described our approaches to entity ranking used to produce the submitted runs. We relied on two strategies, one considering outlinks and their anchor text that can be found at the primary web-pages as entity mentions, and on another one, used in three of four runs, fully relying upon Wikipedia as on the repository of entities. Wikipedia-based approaches outperformed the baseline outlink-base approach. In a few cases, Wikipedia-based techniques failed to show non-zero performance what confirms that the coverage of Wikipedia is limited and can not be used to answer all queries. However, it was possible to retrieve at least one relevant Wikipedia entity among the first 10 in 80% cases (for all topics except the 3rd, 8th, 13th and 16th).

We clearly see two ways to improve the Wikipedia-based entity ranking. First, we need to improve the classification part, since current ontologies do not cover the entire Wikipedia and hence can not serve as a sufficient solution. Second, we need to evaluate the relevance of each entity by analyzing the entity co-occurrence within the smaller context, e.g. at the paragraph or sentence level. Of particular interest is also the task of finding primary/relevant homepages for the given Wikipedia entity.

9. ACKNOWLEDGMENTS

We would like to sincerely thank Claudia Hauff and Djoerd Hiemstra for the help with collection preprocessing.

10. REFERENCES

[1] G. Demartini, A. P. de Vries, T. Iofciu, and J. Zhu. Overview of the inex 2008 entity ranking track. In S. Geva, J. Kamps, and A. Trotman, editors, *INEX*, volume 5631 of *Lecture Notes in Computer Science*, pages 243–252. Springer, 2008.

[2] M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multitheme documents. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 661–670, New York, NY, USA, 2009. ACM.

[3] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, New York, NY, USA, 2007. ACM.

[4] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518, New York, NY, USA, 2008. ACM.