Inferring the Most Important Types of a Query: a Semantic Approach

David Vallet*
Universidad Autónoma de Madrid
Ciudad Universitaria de Cantoblanco
Madrid 28049, Spain
david.vallet@uam.es

Hugo Zaragoza
Yahoo! Research Barcelona
Ocata 1
Barcelona 08003, Spain
hugoz@es.yahoo-inc.com

ABSTRACT

In this paper we present a technique for ranking the most important types or categories for a given query. Rather than trying to find the category of the query, known as query categorization, our approach seeks to find the most important types related to the query results. Not necessarily the query category falls into this ranking of types and therefore our approach can be complementary.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Retrieval Models

General Terms: Algorithms, Experimentation

Keywords: Type Ranking, Entity Ranking, Faceted Search

1. INTRODUCTION

Under the current Web search paradigm, search engines return ordered list of *snippets*, each representing a web page of interest. In the past few years, new forms of search are appearing both in the academic literature and in novel commercial search services. For example, if pages contain metadata (i.e. page categories, types, properties, authors, etc.) this metadata can be used to organise the search results and allow the user to browse or filter based on them. This idea was adopted early by search engines over product databases such as those found in catalog and shopping sites (e.g. www.amazon.com, www.kelkoo.com), travel (e.g. www.opodo.com), etc. Search engines also allow users to select in which category their search falls (e.g. Web, Music, Video). Faceted Search [2] is a recently proposed framework to formalise these approaches. We will use this term loosely here to refer in general to search engines which expose the metadata to the user for browsing. For collections without explicit metadata, faceted search can still be applied by developing automatic classifiers and extractors that process the content of the documents and extract properties.

A crucial and difficult problem in Faceted Search is to choose which are the most important facets for the query. This is a problem of *facet ranking* rather than document ranking. Once the most important facets are determined, they can be used to adapt the presentation of results (changing the display, clustering, providing filters, reranking, etc.)

Entity Retrieval (ER) is a different trend that modifies the traditional search engine paradigm. Entities are phrases with an associated semantic type (e.g. "CITY:San Francisco", "DATE:July 2008"). In ER the result to a user query is not a ranked list of snippets, but rather, a ranked list of entities [1, 3]. Interestingly, Entity Retrieval can provide very useful information to Faceted Search, specially in the case when explicit metadata is not available. By analysing the entities relevant to a query, we can gain information about the types that are most interesting to this query. For example, the query 'New York city' retrieves, in our Wikipedia corpus, both entities related to well-known locations of the city and also entities related to important dates of the city's history. This tells us that the LOCATION and DATE types are important for this query, more so than other types such as PERSON, ORGANIZATION, etc. Similar to the problem of selecting facets, the problem here is to rank types, not entities. Once we select the most important types for a query, we could use this information in a number of ways. For example, using type-specific displays for the most important types (e.g. a map for locations, a timeline for dates) or letting the user filter the results.

We are interested here in the use of entity retrieval for the automatic prediction of the most relevant types for a query. We call this problem *entity type ranking* to differentiate from standard entity ranking. To the best of our knowledge this is the first work that studies this problem.

2. ENTITY AND TYPE RANKING

To study this problem we adopt the entity ranking setting described in [3]. We use the same corpus as them, a snapshot of Wikipedia with named entities automatically extracted: 20.3M occurrences of .8M unique named entities [3]. An entity e is represented by the named entity phrase and its associated type (64 different types and subtypes) t. The entity ranker works as follows. The query is executed against a standard passage retrieval algorithm, which retrieves the 500 most relevant passages and collects all the entities that appear on them. A Kullback-Leibler distance (KLD) based ranking algorithm is applied to the entities[1]:

$$score_q(e) = KLD(P_q(e)||P_c(e)) = P_q(e) * \log(P_q(e)/P_c(e))$$

where $P(e)$ is the prior probability of an entity being in a sentence, $P_q(e)$ and $P_c(e)$ are the maximum likelihood estimates of $P(e)$ computed over the top 500 retrieved sentences

*during a short internship at Yahoo! Research Barcelona

SIGIR'08, July 20–24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

[1]This improved slightly (3% AvgPrec on average) over the ranking methods proposed in [3].

Table 1: Example of queries and associated types

Query	Query Type	Type 1	Type 2	Type 3
Australia	Country	Location	Date	Organization
Hanseatic League	Organization	Location	Date	Organization
Paris Dakar	Event	Location	Person	Vehicle

for query q and over the entire corpus respectively.

The entity ranker system returns for a query a ranked list of entities $E(q) = e_1, e_2, \ldots, e_{n_q}$, ordered by their decreasing score. The *type ranking system* takes this as an input and needs to produce a ranked list of entity types $T(q) = t_1, t_2, \ldots, t_{n'_q}$ from most to least important types related to the query. The family of type ranking functions that we have experimented with can be expressed as:

$$score(t,q) = \sum_{i=1}^{n_q} \begin{cases} w_q(t,i) & \text{if } type(i) = t \\ 0 & \text{otherwise} \end{cases}$$

where $w_q(t,i)$ is a weighting function for that type and that query at the given position i of the ranked entity list E. We tried a number of weighting functions; we report here the four which seem more interesting to us [2]:

- $w_q(t,i) = count_q(t,i) := 1$
- $w_q(t,i) = score_q(t,i) := score_q(e_i)$
- $w_q(t,i) = pos_q(t,i) := (n_q - i)$
- $w_q(t,i) = pos_q^2(t,i) := (n_q - i)^2$

We also tried using these weighting functions only on the top k entities in E.

3. EXPERIMENTS AND CONCLUSION

The weighting functions were evaluated with 50 queries. The type relevance assessments for these queries were created by 1) launching the queries with the entity retrieval system, 2) making a relevance assessment of the returned entities and 3) ordering the list of types with the highest percentage of entities judged as relevant. Table 1 shows an example of queries, their type (or category), and the three top most important types, inferred from the assessments as described above. The query type does not always coincide with the top most important types. For instance, while the Hanseatic League is an organization, the most important types are the locations (countries, cities) that were part of this trading alliance, the significant dates of the alliance's history and finally other related organizations. As evaluation metrics we used NDCG values (with gain values of 10,5,2) and P@N values. The precision values try to evaluate how effective would be the system on selecting a set of relevant types given a query. Is thus defined as the percentage of relevant result types up to position N that belong to the top N relevant types from the assessments.

Table 2 shows the results of the evaluation. Values between parenthesis are the results using only the top $k = 70$ entity results from $E(q)$. This value of k led to the maximum value over a subset of 20 queries for all type ranking weighting functions.

[2]We also tried other polynomial and exponential discounting functions, without improvements.

Table 2: Average NDCG and P@N

	NDCG	P@1	P@2	P@3
count	0.651(0.766)	0.480(0.660)	0.540(0.630)	0.607(0.693)
score	0.671(0.628)	0.520(0.460)	0.560(0.490)	0.593(0.573)
pos	0.678(0.769)	0.480(0.660)	0.590(0.640)	0.640(0.693)
pos^2	0.733(0.774)	0.640(0.680)	0.610(0.660)	0.627(0.700)

The main difference between the four proposed approaches is the importance that the weighting function gives to the top result types. *count* gives the same weight regardless of the position, and even being the most naive approach, it still achieves a considerable performance. The *score* function gives slightly more importance to the top results, as their score values are higher, but its improvement is marginal. The weighting functions *pos* and pos^2 (to a higher degree) give more importance to the first results. *pos* slightly improves the baseline approach, whereas pos^2 yields a greater improvement: 13% on NDCG 33% on P@1. This suggests that the entities with the relevant types are more frequent on the upper positions of the results sets of our entity retrieval system. The latter function seems to adapt better to this distribution. This hypothesis is further validated by examining the top $k = 70$ modification results, showing improvements ranging from a 18% NDCG on the baseline approach, to a 13% and 6% on the *pos* and pos^2 approaches, which already give more importance to the top positions.

4. CONCLUSION

In this work, we propose the task of entity type ranking, and present a method to predict the more important types relevant to a query in an informational search task. We do this by making use of entity extraction and entity ranking systems. The proposed methods can achieve up to 70% precision on the three top inferred types. This can have direct application to faceted search systems, specially in informational search and with corpora where metadata needs to be extracted from the documents.

5. ACKNOWLEDGEMENTS

This research was partially supported by the European Commission under contract FP6-027685 MESH. The expressed content is the view of the authors but not necessarily the view of the MESH project as a whole.

6. REFERENCES

[1] A. P. de Vries, J. A. Thom, A. M. Vercoustre, N. Craswell, and M. Lalmas. Inex 2007 entity ranking track guidelines. In *INEX 2007 Workshop preproceedings*, pages 481–486, 2007.

[2] M. Tvarozek and M. Bielikova. Adaptive faceted browser for navigation in open information spaces. In *WWW '07*, pages 1311–1312, New York, NY, USA, 2007. ACM.

[3] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on wikipedia. In *CIKM '07*, pages 1015–1018, New York, NY, USA, 2007. ACM.