An Evaluation of Entity and Frequency Based Query Completion Methods

ABSTRACT

We present a semantic approach to suggesting query completions which leverages entity and type information. When compared to a frequency-based approach, we show that such information mostly helps rare queries.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query Formulation

General Terms

Algorithms, Experimentation, Measurement

Keywords

Query Completion, Query Log Analysis, Web Search

1. INTRODUCTION

From the days of boolean search on library catalogues, users have reformulated their queries after an inspection of initial search results. Traditional information retrieval studied this in frameworks such as query expansion, relevance feedback, interactive retrieval, etc. These methods mostly exploit document contents because that is typically all information that is available. The situation is very different in web search engines because of the large amounts of users whose queries are collected in *query logs.* Query logs reflect how large numbers of users express their queries and can be a rich source of information when optimizing search results [3] or determining query suggestions [1, 4, 5].

In this paper we study a special case of query suggestion: query completion, which aims to help users complete their queries. In particular, we are interested in comparing a commonly adopted frequency-based approach with methods that exploit an understanding of the type of entities in queries. Our intuition is that completion for rare queries can be improved by understanding the type of entity being sought. For example, if we know that "LX354" is a kind of digital camera, we can generate *sensible* completions by choosing them from the set of completions used with other digital cameras. Besides suggesting queries, the obtained completions can also function as facets for faceted browsing or as input for ontology engineering since they represent query refinements common to a class of entities. In this paper, we address the following questions: (i) How can we recognize entities and their types in queries? (ii) How can we rank possible completions given an entity type? (iii) How can our methods be evaluated and how do they perform? To address (iii), we propose a novel method which evaluates the prediction of real web queries. We show that a purely frequency-based approach without any entity type information works quite well for more frequent queries, but is surpassed by type-based methods for rare queries.

2. METHODOLOGY

In this section we introduce our notations and proposed type-based methods. We assume queries can be decomposed in an entity part e and a completion part f and that entities can be assigned a type T. Table 1 shows some examples. In case the query contains both a prefix and suffix, we treat it as two separate queries. Given a set of such queries, we determine the matrix $N = (n_{ef})_{e,f}$, where n_{ef} is the number of times we see f with e. By grouping all entities of a certain type we can, for example, compute $n_{Tf} := \sum_{e \in T} n_{ef}$ which is the number of times we see completion f with an entity of type T. Using N, we can readily estimate probabilities such as $P(f)$, $P(f|e)$, $P(f|T)$, and $P(e|f,T)$.

Suggestion Methods

Imagine that a user is typing a query and we recognize what she has typed so far as an entity with a corresponding type. The most naive approach would be to suggest the most frequent completions for the current entity (**M0**): $score_{M0}(f, e) = P(f|e)$. Given an infinite amount of data this should suffice. However, it will probably fail for rare entities since we will have no or very few completions for these. For this reason we turn to the entity type and smooth the entity distribution with the type distribution.

M1 looks at the most likely completion for the current type: $score_{M1}(f, T) = P(f|T)$. Another desirable property a completion should have, is being rare over all types. **M2** rewards such completions: $score_{M2}(f, T) = \frac{P(f|T)}{P(f)}$. Another intuition is that completions which are frequent as well as evenly distributed among the entities in the type should be rewarded (**M3**): $score_{M3}(f, T) = G(f|T) = \left(\prod_{e \in T} n_{ef}\right)^{1/|T|}$. The final method (**M4**) only considers the distribution of completions within the type: $score_{M4}(f, T) = H(\theta_{f|T})$, where H is the entropy of the multinomial $\theta_{f|T} = (P(e|f, T))_{e \in T}$.

To detect entities in queries, we use a *dictionary* (i.e. a list of all the entities of each type) and match the largest substring common to the query and the dictionary. Dictionaries can be built from taxonomies which list entities of different types. We derive a taxonomy from Wikipedia categories and assume that every category constitutes a type (e.g. for the entity Madonna: American actor-singers, American dancers,

Query	Entity	Completion	Type
aspirin side effects	ASPIRIN	+*side effects*	Anti-inflammatory drugs
how to take ibuprofen	IBUPROFEN	–*how to take*	Anti-inflammatory drugs
britney spears video	BRITNEY SPEARS	+*video*	American film actors

Table 1: Example queries, extracted entities, completions, and types.

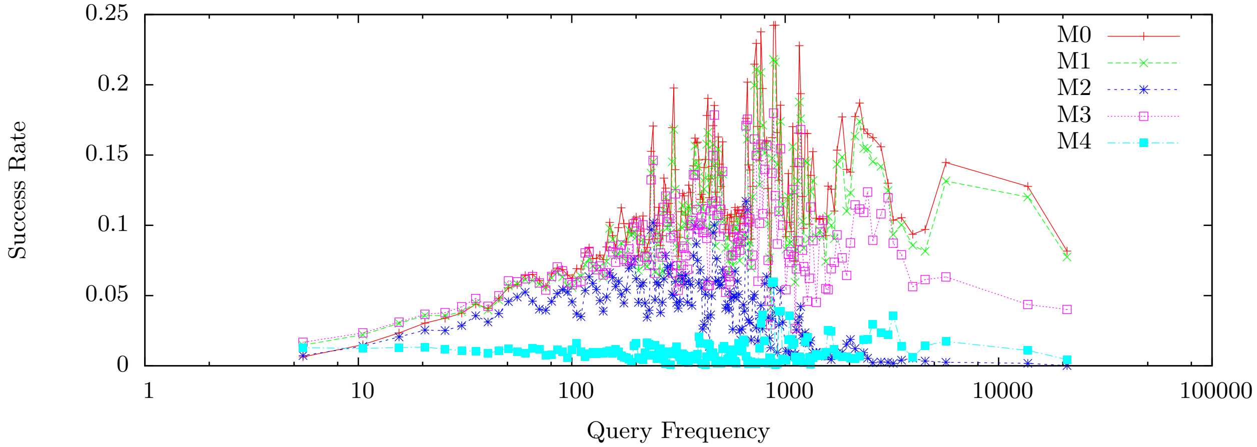

Figure 1: Success rate for queries. The x-axis shows the queries which are binned by frequency of occurrence.

etc.). Further, every article listed in a category constitutes an entity of that type[1]. Wikipedia is very large with a broad coverage and it defines many possible entities (our version has 1,970,637 entities and 269,393 types). Wikipedia entries can belong to many categories (e.g. 34 for Madonna). So, we need a method to choose the *best* entity type. This is a challenging research question in itself; trivial methods such as choosing the most frequent or rarest type did not work well. Instead, we apply M1 on the training data and evaluate the performance of all possible types of each entity. We then choose the type that led to the best performance on the training set. For entities not present in the training set we select the type with the most entities.

3. EXPERIMENTS AND RESULTS

For evaluation, we compare the highest scoring completions of the various methods with the actual observed remainder of the queries in the test set. We use 6 consecutive days of query logs which we split equally into a training and a test set. We analyze each query and if it contains an entity we keep it. This results in 1,681,753 queries for training and 1,644,033 for testing. For each query, we compute the top $K = 10$ completions predicted by each method. The correct completion for that query is the one typed by the user. We are interested in two evaluation measures: (i) Success Rate @ K (SR), i.e. whether the completion is correctly predicted and (ii) Mean Reciprocal Rank @ K (MRR), i.e. the mean of the inverse of the ranks at which the completion was found, up to K. Table 2 shows the results over all test queries. As is clear from the low absolute scores, the task of suggesting the correct completion is a difficult one. The highest obtained MRR is 0.178 for M0 with queries that occur around 1000 times. M0 outperforms the type-based methods on almost all queries and measures. However, as indicated by Figure 1, the type-based methods M1 and M3 perform slightly better than M0 for less frequent queries (occurring 40 times or less and making up 12.7% of the total query volume). For other queries, M0 outperforms all other methods although the difference with M1 is usually small. The reason for the drop in performance for frequently occurring queries is that these mostly consist of entities such as "in", "to", and "the" (which are Wikipedia entries, but do not seem sensible entities).

	M0	M1	M2	M3	M4
MRR	0.081	0.068	0.014	0.046	0.006
SR	0.118	0.104	0.041	0.088	0.010

Table 2: Aggregated results over all queries.

4. CONCLUSION AND FUTURE WORK

We have presented a semantic approach to suggesting query completions, which leverages entity and type information. We evaluate various methods and compare them to a purely frequency-based approach. The results show that type-based information mostly helps queries which do not occur frequently. Our current work is only the first step towards a thorough evaluation methodology for this task. Future work includes performing a user study to obtain an indication of the usability of the type-based suggestions. We also intend to look at different ways of defining entity types and the mapping of entities to types.

5. REFERENCES

[1] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *SIGIR '03*, 2003.

[2] J. Bai and J.-Y. Nie. Adapting information retrieval to query contexts. *IPM*, 44(6):1901–1922, 2008.

[3] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, 2002.

[4] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW '06*, 2006.

[5] X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. In *CIKM '08*, 2008.

[1] We remove any disambiguation part in the entry title. This has the adverse effect of introducing noise, e.g. collapsing Madonna (art) and Madonna (entertainer). Disambiguating such queries is beyond the scope of the current work but could, e.g., be achieved by leveraging a user's history [2].