A Query Construction Service for Large-Scale Web Search Engines

Ioannis Papadakis
Department of Archives and Library Sciences
Ionian University, GREECE
papadakis@ionio.gr

Michalis Stefanidakis
Computer Science Department
Ionian University, GREECE
mistral@ionio.gr

Sofia Stamou
Computer Engineering and Informatics Dept.
Patras University, GREECE
stamou@ceid.upatras.gr

Ioannis Andreou
Digital Systems Department
University of Piraeus, GREECE
gandreou@unipi.gr

Abstract

The most popular way for finding information on the web is go to a large-scale search engine and submit a query. Despite their wide usage, large-scale search engines are not always effective in tracing the best possible information for the user needs. There are times when web searchers spend too much time searching over large-scale web search engines before obtaining the anticipated results. When (if) they eventually find the sought information, they often realize that their successful queries are significantly different from their initial one. In this paper, we introduce a query construction service for assisting web information seekers specify precise and unambiguous queries over large-scale web search engines. The proposed service leverages the collective knowledge encapsulated mainly in the Wikipedia corpus and provides an intuitive GUI via which web users can determine the semantic orientation of their searches before these are executed by the desired engine.

1. Introduction

Currently, large-scale web search engines are the predominant mean for accessing the flourishing data that is available on the web. One thing that makes search engines so popular is that they enable users query the web in an intuitive yet simple manner, i.e. by submitting a few keywords to the engine's search box. Despite the intended simplicity associated with querying the web via a large-scale search engine, there are times when web searchers spend too much time reformulating queries, without being able to satisfy their information needs. Search engines provide little help to users with vague knowledge of the terminology employed within relevant documents. Even if searchers succeed in locating the information sought, they often realize that their successful queries differ significantly from their initial query.

In this paper, we propose a query construction service that extends the functionality of the traditional search box and acts as an intermediate layer between searchers and large-scale web search engines. Specifically, the service's search box incorporates auto-suggest functionality that offers query suggestions based on the semantic information of an underlying ontology. Upon selection of a suggested query by the user, the latter is provided with information about the semantics of his selected query. Semantic information is visualized as a conceptual ontology whose nodes represent concepts and whose labeled links represent the semantic relation that connects concepts together. By traversing the ontology, the user can improve his query and submit it for search. The ontology contains information driving mainly from Wikipedia[1] and is exposed to the searcher through an interactive, ontology-browsing GUI.

The rest of the paper is organized as follows. We begin our discussion with an overview of the different search modes that web users employ and outline a number of approaches that have been proposed for improving the users' search experience. In Section 3, we introduce our query construction service and in Section 4, we provide several snapshots of our service's GUI in order to illustrate the functionality and intuitiveness of our proposed method. In Section 5, we discuss the main advantages of our proposed service and we conclude the paper in Section 6.

2. Preliminaries and Background Works

2.1. Web Querying Behaviors

It is common knowledge that searchers do not employ a standard behavior when querying the web. This is essentially because people have different backgrounds and varying needs and thus they make their

[1] www.wikipedia.org/

query selections based on different criteria and underlying knowledge. Currently, there exist a number of studies (e.g. [3], [6]) that try to elucidate the different search modes that web users employ. In this direction, [5] identified four intersecting information seeking modes: (i) the known-item, (ii) the exploratory mode, (iii) the don't know what I need to know and (iv) the re-finding mode. Given that the aim of our study is to help users formulate good queries in various information seeking modes, we rely upon the research suggested in [5].

In particular, the **known-item** search mode adheres when the user has a specific information need and is capable of picking the right keywords for specifying his query. Under the known-item search, any difficulties that search engines encounter with respect to answering known-item queries emerge from the intrinsic nature of natural languages, as we discuss next.

The **exploratory** search mode is employed when the user has a specific information need but is not sure how to express it in a set of keywords. Under this search mode, the challenge that search engines encounter is how to assist users formulate intention-descriptive queries.

The **don't know what I need to know** mode refers to the situation that a user submits a query without a specific goal in mind. Such searches might occur in complex or unknown domains (i.e. legal, medical) as well as in case the user's need is to get an update of what is on the web about his query. The paradox of this search mode is that neither the user nor the engine are able to resolve the intention of the query without the assistance of some external resource, e.g. pages retrieved for an initial query [4]. Thus, the greatest challenge is how to help users crystallize their search goals at query time.

Finally, the **re-find** mode is encountered when the user queries the engine in order to find information that he has already seen in a previous search. Such a mode can be dealt with outside the context of the search engine. Therefore, it will not concern us further in this paper.

2.2. Search Engines' Query Handling

Although information seekers employ different strategies when querying the web, large-scale search engines' main concern is just to find the most efficient way to rank search results. However, without any help from the search engine at query construction time, searchers that do not know exactly what they are looking for and/or how to express it as a search query, are most likely to issue a misleading query that will eventually result in the retrieval of perfectly ranked, irrelevant documents. In addition, studies have shown that a significant fraction of search queries are underspecified and contain only a few words. Short queries, being marginally informative of the users' search intentions, result oftentimes to the retrieval of search results that might not satisfy the user information needs. To make things worse, queries might be ambiguous or polysemous; using identical terms to represent distinct information needs, which constitutes the retrieval of relevant data arduous. Evidently, as users become more dependent on web data to find information about a subject of interest, there is an ever-increasing need that we equip search engines with modules that can assist information seekers select queries that express their varying search intentions in a distinguishable by the engine manner.

2.3. Query Selection for Improved Searches

In an effort to improve the users search experience, Google's search wiki[2] and Yahoo's SearchMonkey[3] approaches take advantage of their email services for authenticating their users and consequently log their personal search tactics in order to provide them with personalized search results. They both employ auto-suggest functionality within the search box and they both anticipate explicit feedback from the searchers during their searching process. Upon addressing a query to the search engine, Google's approach presents a list of pairs. Each pair contains a suggested query constructed by the search engine together with the corresponding first result. This way, users have the chance to disambiguate their initial query by choosing the suggestion that best matches their information needs. On the contrary, Yahoo's approach requires explicit feedback from the searchers during their searching process. Whether they provide enough motives for the web searchers to spend extra time in providing such feedback, still remains to be seen.

3. Query Construction Service

The proposed service consists of a client and a server. The client utilizes Ajax technology and serves towards augmenting user typed keywords with information from DBpedia [1]. The server is written in python and uses the *Twisted http server framework*[4] to fulfill automated client-side requests, by retrieving data related to the user needs from the underling DBpedia-based ontology. Retrieval queries are eventually trans-

[2] Google's search wiki, http://www.google.com/psearch

[3] Searchmonkey, http://developer.yahoo.com/searchmonkey

[4] http://twistedmatrix.com

formed into relational SQL-select statements since the ontology is stored into a MySQL database.

So far, our service utilizes the following DBpedia datasets: (i) the Wikipedia articles, (ii) the list of disambiguations that Wikipedia encodes for connecting generic articles to their specific interpretations, (iii) the categories under which the Wikipedia articles are classified, (iv) the WordNet classes to which Wikipedia articles correspond and which mainly pertain to the synset name that represents the entities' corresponding properties and features and (v) the articles' infobox datasets that contain semantically rich key-value pair of properties about the considered articles. Table 1 summarizes the DBpedia data that we explored.

Table 1. Statistics on the Wikipedia harvested data

Collection period: January 2009	
Dataset	**Value**
Wikipedia articles	2,866,994
Disambiguation entries	226,978
Categories	339,112
WordNet classes	124
Articles linked to WordNet classes	497,797
Infobox records	19,230,789

Relying on the above dataset, we organized it into an ontological scheme as shown in Fig. 1

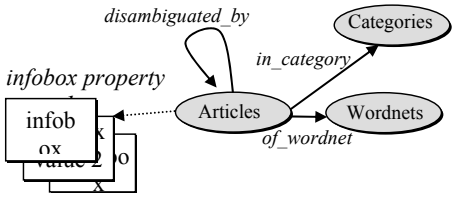

Figure 1. Ontology Schema.

The class *Articles* contains all the Wikipedia article references organized as class instances. The class *Categories* is employed to host the respective categories of the Wikipedia articles. *WordNet classes* store all the possible types of the article entities. The article disambiguations are expressed as a reflexive '*disambiguated_by*' relation. Similarly, the '*in_category*' relation is employed to link articles to their belonging categories. Furthermore, the articles that are associated with WordNet classes are linked to their respective entity type encoded in WordNet via the '*of_WordNet*' relation. Finally, the Wikipedia infoboxes of name-value pairs of properties are ontologically expressed as datatype properties of their corresponding article instances (sketched as dotted arrow in Fig. 1).

4. GUI for Formulating Queries

The principle upon which the GUI design took place is that it should be interactive, inductive, easy to use and fast to execute. Having such requirements in mind and based on the work of [2] on ontology visualization, we designed the GUI as follows. Upon typing a few characters of a query, the provided search box suggests a number of strings that can be attached to the typed tokens in order to complete them. The auto-complete suggestions are leveraged from the titles of the Wikipedia articles that our service encapsulates. In case the user does not wish to employ any of the suggested query alternatives, he can ignore the suggestions and search with his self-selected keywords. On the other hand, if the user selects a suggested alternative; an HTTP-GET request is addressed to the server aiming at extracting semantic information for the selected query suggestion. Semantic information pertains to (i) the query disambiguations (possibly) grouped by the WordNet classes to which disambiguations belong, (ii) the Wikipedia categories associated with each of the suggestions and (iii) key-value pairs harvested from the Wikipedia infoboxes

Query Disambiguation. Query disambiguation is performed in one or two steps: at first, upon selecting the 'disambiguated by' box, the user receives a list of all the corresponding disambiguations that match his selected suggestion (Fig. 2).

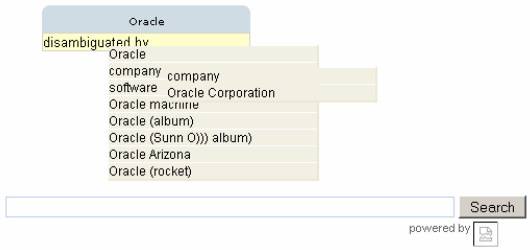

Figure 2. Query Disambiguations.

Such disambiguations could be grouped by a WordNet class, provided they share a common WordNet meaning. In such case, upon selecting the corresponding WordNet label, a second-level disambiguation list appears. By selecting either one of the first- or second-level disambiguations, a new box containing the disambiguated entity is sketched at the right, which is connected to the previous box with a line labeled 'disambiguated by'. Simultaneously, a search query that consists of keywords deriving from the two box titles (elimination of duplicates is applied) is addressed to the underlying large-scale search engine.

Categories. In case the selected suggestion (from the search box) is associated with Wikipedia categories, the label 'in category' appears on the interface

and a similar process is initiated. The searcher is prompted to select the 'in category' relation in order to find the category that best matches his intended query semantics. Upon category selection, a new box containing the selected category is sketched at the right of the interface and is connected to the previous box with a line labeled 'in category' (Fig. 3).

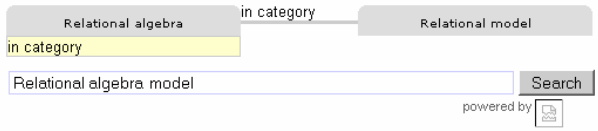

Figure 3. Selected Category for the Disambiguated Query.

Simultaneously, a search query that consists of keywords deriving from the two box titles (elimination of duplicates is applied) is addressed to the underlying search engine.

Infoboxes. Finally, if the selected suggestion (from the search box) is associated with Wikipedia infoboxes, these are displayed as labels beneath the query inside the box. Similarly to the 'disambiguated by' and 'in category' options, the user is prompted to select a key in order to obtain the corresponding values. Upon the selection of a value, a new box containing the selection is sketched at the right and connected to the previously selected box using a line labeled with the key name of the selection. At the same time a search query consisting of the keywords deriving from the two box titles is addressed to the underlying search engine. Our query construction service has so far been integrated with two major web search engines (Google and Yahoo) and is accessibleonline[5]. Thus, we believe that the integration is doable for any search engine that gives programmable access to its search box.

5. Discussion

The main advantage of our service is that the user always maintains control of the query construction process. Moreover, by employing our service, the searcher is instantly acquainted with query terms that would otherwise take him a lot of time to gather by exhaustively scanning the results of his initially vague query. Furthermore, the simplicity of the underlying architecture not only renders the proposed service scalable to future enhancements with more semantically-rich datasets, but also guaranties its rapid execution time. The above features are very important for large-scale web search engines where time and space play a crucial role for their prosperity. The employment of common web widgets such as the auto-suggest box and clickable divisions ( ) as well as the absence of semantic web terminology from the GUI, renders the proposed service fast to learn and easy to use. Finally, we should mention that if there is no information in Wikipedia about the user typed terms, the query is transparently forwarded to the search engine that our system integrates. Therefore, the worst case scenario in the search process is that the user does not get any help from the service, but still his query is automatically submitted to the underlying engine for search.

6. Conclusion

In this paper, we introduced a novel query construction service that has been seamlessly integrated with large-scale web search engines in an attempt to assist information seekers specify precise and unambiguous queries. The proposed service relies upon the semantic information stored in DBpedia, which it organizes into an ontology and enables users understand the semantic orientation of their search keywords before these are actually issued for search. To demonstrate the provided functionality, we have implemented an interactive, non-intrusive and easy-to-use query construction service via which users obtain information about the semantics of their search terms as well as alternative wordings for verbalizing their search intentions. Semantic information is gradually provided to the users upon request and helps them crystallize their search pursuits progressively.

7. References

[1] Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R. and Ives, Z. 2007. DBpedia: a nucleus for a web of open data. In Proc.of the 6th. Semantic Web Conference.

[2] Papadakis I., Stefanidakis M. 2008. Visualizing ontologies on the web, New Directions in Intelligent Interactive Multimedia, Studies in Computational Intelligence, vol. 142, Springer Verlag, pp. 303-311.

[3] Broder, A. 2002. A taxonomy of web search. SIGIR Forum, 36(2).

[4] Milne, D.N., Witten, I.H. and Nichols, D. 2007. A knowledge-based search engine powered by Wikipedia. In Proc. of the 16th Conf. on Knowledge Management, pp.445-454.

[5] Spencer, D. 2006. Four models of seeking information and how to design for them boxes and arrows. Available at: http://www.boxesandarrows.com/view/four_modes_of_seeking_information_and_how_to_design_for_them

[6] Spink, A., Park, M., Jansen, B. and Pedersen, O. 2006. Multitasking during web search session. In Information Processing and Management, 42(5): 1366-1378.

[5] http://195.251.111.53/snh/entry/index.html for Google and http://195.251.111.53/snh/entry/index2.html for Yahoo.