J Internet Serv Appl (2012) 3:255–268
DOI 10.1007/s13174-012-0068-9

ORIGINAL PAPER

Semantifying queries over large-scale Web search engines

Ioannis Papadakis · Michalis Stefanidakis · Sofia Stamou · Ioannis Andreou

Received: 24 June 2011 / Accepted: 20 August 2012 / Published online: 12 September 2012

Abstract In many situations, searching the web is synonymous to information seeking. Currently, web search engines are the most popular vehicle via which people get access to the web. Their popularity is partially due to the intrinsic way that people interact with them, i.e., by typing some keywords to the corresponding input box. Despite their popularity, search engines often fail to satisfy certain information needs, especially when the latter are haze and poorly articulated. In this paper, we focus on the occasions when large-scale web search engines find it difficult to cope with specific information-seeking behaviors and we accordingly introduce a query construction service that is targeted towards the solution of this problem. The proposed service leverages information coming from various DBpedia datasets and provides an intuitive GUI via which searchers determine the semantic orientation of their queries before these are addressed to the underlying search engine. The evaluation of the query construction service justifies the motive of this paper and indicates that it can considerably improve the searchers' querying ability when search engines fail to provide adequate help.

I. Papadakis (✉) · S. Stamou
Department of Archives and Library Science, Ionian University,
Ioannou Theotoki 72, 49100 Corfu, Greece
e-mail: ipapadakis@gmail.com; papadakis@ionio.gr

S. Stamou
e-mail: stamou@ionio.gr

M. Stefanidakis
Department of Informatics, Ionian University,
7 Tsirigoti Square, 49100 Corfu, Greece
e-mail: mistral@ionio.gr

I. Andreou
Department of Informatics, University of Piraeus,
Karaoli and Dimitriou 80, 18534 Piraeus, Greece
e-mail: ioannis.andreou@gmail.com

Keywords Wikipedia · DBpedia · Search engines · GUI design · Semantic web

1 Introduction

Currently, large-scale web search engines are the obvious choice 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 input 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, before being able to satisfy their information needs. For example, search engines provide little help to searchers with vague knowledge of the terminology employed within relevant documents. Even when searchers succeed in locating the information sought, they often realize that their successful queries differ significantly in terms of vocabulary from their initial search request.

In this paper, we introduce a query construction service that resides on top of large-scale web search engines and aims at assisting information seekers formulate queries that are expressive of their search intentions. By doing so, we implicitly improve search engines' capability of understanding the user information needs and accordingly of serving their queries. As discussed in the paper, the proposed approach is particularly useful in specific information-seeking modes, where web searchers are unable to address accurate and specific queries to large-scale search engines. The motive of this work is to bridge the semantic gap between the initial and the resulting query of a web search session consisting of several interactions between the searcher and the respective web search engine.

The proposed query construction service acts as an intermediate layer between searchers and large-scale web search engines. From the GUI's perspective, such a layer is realized by extending the functionality of the search engine's input box. Currently, leading search engines on the web such as Google and Yahoo! offer query suggestions that begin with the same letters a searcher types in the input box. Such suggestions are usually ranked according to their popularity. Our method suggests query alternatives based on the titles of Wikipedia[1] articles (acting as queries) that are provided by the intermediate layer. Then, upon selection of a suggestion, the searcher is prompted to a simple yet intuitive and interactive interface that assists him reconstruct his query based on the semantics of the respective Wikipedia article (acting as the initial search query).

Given the difficulty of evaluating a service that is practically addressed to the entire web population, we performed a qualitative analysis and a human survey in order to receive some indicative feedback about the user perceived performance of our method. The results of the evaluation are very encouraging, regarding both the motive of this work and the effectiveness of the proposed solution. More specifically, according to our survey, there are cases in which large-scale web search engines have limited ability in providing sufficient help to their users while articulating their information needs as search keywords. Conversely, our method is more successful in driving users to the selection of terms that are accurate in specifying their information needs.

The rest of the paper is organized as follows. We begin our discussion with an overview of the different search modes that web information seekers employ. Then, we present the main difficulties associated with querying the web and discuss a number of methods that have been proposed for addressing such difficulties. In Sect. 3, we introduce our query construction service. Specifically, we describe how we leverage knowledge from Wikipedia not only to provide searchers with query suggestions, but also to help them perceive the semantics of their selected query terms. In Sect. 4, we discuss the proposed approach, and in Sect. 5 we describe the evaluation we carried out for assessing the usefulness of the proposed service while searching the web. Finally, we conclude our work in Sect. 6.

2 Preliminaries and related work

In this section, we describe the different search behaviors that users adopt when querying the web, to illustrate how these affect both the engines' retrieval performance and the users' search experience. Then, we outline the current search paradigm that search engines support to address the corresponding difficulties. Finally, we discuss several methods that have been proposed for assisting web searchers specify good queries.

2.1 Web querying behaviors

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 query selections based on different criteria and underlying knowledge. A number of existing studies have tried to elucidate the different search modes that web searchers employ. For instance, Carmel et al. [13] propose a query taxonomy for classifying search intentions. Based on this taxonomy, researchers [22] experimentally evaluated the different goals associated with queries and showed that the query intention for particular types of searches can be safely predicted. In a similar direction, Spencer [25] identified four intersecting information-seeking modes, namely: (i) known item, (ii) exploratory, (iii) don't know what I need to know and (iv) re-find. In the course of our study, we built a query construction service that assists searchers who engage in the above seeking modes. Before delving into the details of the service, we present the characteristics of the above search modes, as determined by Spencer.

In particular, the `known-item` search mode is used when searchers have a specific information need and they are capable of picking suitable keywords for verbalizing this need. Under the known-item search, retrieval effectiveness is heavily dependent on the lexico-semantic properties of the query terms and has little to do with the searchers' competence in verbalizing their search pursuit. Any difficulties that search engines encounter with respect to answering known-item queries emerge from the intrinsic nature of natural languages. Thus, search queries expressed as consecutive terms suffer from the possible polysemy of the words that constitute the search query. Polysemy occurs when a word has more than one sense [18]. A query intended to elicit information resources relevant to one sense of a polysemous word may elicit unwanted information resources relevant to other senses of that word. Moreover, search engines have to overcome another feature of spoken languages: synonymy of words. Synonymy occurs when two or more words share the same meaning [18]. The magnitude of synonymy's influence to search engines can be further realized by taking under consideration the fact that the probability of two persons using the same term in describing the same thing is less than 20 % [7].

When dealing with the `exploratory` information-seeking mode, search engines have to deal with the fact that searchers are not always able to properly formulate certain queries, since they do not know how to phrase such queries. Thus, the searchers' inability to express what they are after

[1] http://wikipedia.org

within the environment of the search engine forces them to perform an initial search and exhaustively run through the search results in order to learn about the corresponding domain [8] and eventually get some ideas about relevant terms and their synonyms. This is a rather tedious process that involves running through a lot of useless information to find comparatively smaller pieces of useful information.

The `don't know what I need to know` information-seeking mode consists of searches occurring in a complex and/or unknown domain (e.g., in legal, medical, financial) as well as searches addressing the need of keeping up to date. As stated in [19], before the submission of their initial query, searchers are asked to confront the paradox of describing something they do not know without any help from the search engine. Similarly to the exploratory mode, the respective workflow implies that information seekers have to select an appropriate first query that acts as a start point for their search and exhaustively run through the results to learn about the domain and select some useful terms that will help them refine the initial query, participating in this way in an incremental feedback cycle [3].

Finally, the `re-find` mode is encountered when the searchers employ the search engine to find information that they have already seen in a previous search. Such searches can be addressed outside the search engine's context, and thus they will not concern us further.

2.2 Search engines' query handling

Although information seekers employ different strategies when querying the web, large-scale search engines adopt a common approach for serving queries: they look for indexed documents that contain the query terms. Their main concern is dealing with the presentation and ranking of search results rather than assisting searchers specify intention-descriptive queries. But, treating all searches in a uniform manner might harm retrieval performance, especially when dealing with short queries whose intentions are hazy and under-specified. Evidently, as users become more dependent on the web to find information about a subject of interest, there is an ever-increasing need that search engines are enhanced with modules that can assist information seekers select queries that express their varying search intentions in a distinguishable manner by the engine.

2.3 Query selection for improved searches

To assist web searchers overcome the difficulties associated with specifying their information needs on various topics via a limited vocabulary, many techniques have been proposed, e.g., search personalization, relevance feedback, human-powered search, query refinement, clustering, etc.

2.3.1 Search personalization

Search personalization is the process of incorporating information about the user needs in the query processing phase. One approach to personalization is to have users describe their general search interests, which are stored as personal profiles [21]. Recent search personalization approaches on the web involve integration with some kind of external semantic structure, to identify the context of each search session [2,15,24].

More specifically, the authors of [2] describe a profile representation using Internet domain features extracted from URLs. In [24], an effort is made to model the user context as an ontological profile by assigning implicitly derived interest scores to existing concepts deriving from the Open Directory Project (ODP) ontology.[2] Another search personalization technique based on the ODP ontology is defined in [15]. More specifically, a user profile is built by accumulating graph-based query profiles in the same search session. In contrast to [2], the user profile is represented as a graph of the most relevant concepts of an ontology in a specific search session and not as an instance of the entire ontology.

2.3.2 Relevance feedback

Another approach employs relevance feedback. Relevance feedback dictates that queries are reformulated, based on previously retrieved relevant and non-relevant information [23]. Such a technique provides a controlled query alteration process that is designed to emphasize some terms and to deemphasize others, as required in particular search environments. Relevance feedback cannot be easily applied to large-scale web search engines, where authentication is difficult to impose and diversity prevails. Moreover, as noted in [12], it is overambitious to expect searchers to voluntarily provide feedback to the overall information-seeking process without proper motivation. Even in the case of automatic (i.e., blind, without authentication) relevance feedback, where terms from the top few information resources returned are automatically fed back into the query [11], success is by no means self-evident.

2.3.3 Human-powered search engines

Recently, personalization moved towards user community-based information [10], examples of which are the so-called "human-powered" search engines. The phrase "human-powered" refers to a search engine which has its results list affected by human intervention, usually by people rating individual results further up or further down [12]. The rationale behind human-powered search engines is the fact

[2] http://www.dmoz.org

that machines are excellent in executing code very fast but they have no real intelligence, they do not share, they cannot judge and they have no appreciation. So, a search engine that enables its users to determine the position of an information resource in a search results list (capturing this way "the wisdom of the crowd" [27]) will always be better than the most algorithmically efficient large-scale search engine.

In this line of thought, a number of human-powered search engines have emerged. Some of them rely exclusively on their users to build search results lists (e.g., Stumpedia[3]), but most of them do not try to "reinvent the wheel". Instead, they behave as hybrid search engines, applying human intervention to re-rank results that are initially machine generated. Anoox[4] and iRazoo[5] for example, depend on a voting system in order to affect the position of an information resource within a search results list. According to the founders of such systems, people's opinion always outweighs machine's algorithms.

Another kind of human-powered search engine (also called social search engines) tries to improve its quality by applying social networking logic to the underlying workflow. For example, MySidekick[6] is a human-powered search engine that allows people to find and submit information resources. Such resources are automatically tagged with terms that have been used during the search session. The information resources are then anonymously shared within the MySidekick community.

Finally, Wikia[7] is a human-powered search engine that is based on the concept of Wikipedia and founded by its owner. According to Wikia, users are able to collaboratively edit, annotate, comment, delete and expand their search results.

As a common ground, it is argued that human-powered search engines suffer from the fact that they are still just as easy to "game" as more traditional engines [12] and from the fact that they have to persuade their users to provide feedback (implicit or explicit) in order to succeed.

2.3.4 Search results manipulation

Another line of research focuses on the visualization of search results. In this context, Yippy[8] (formerly known as Clusty) and the work presented in [29] are both efforts to aid web searchers by organizing search results in topical clusters. Both approaches add a sidebar containing clusters next to the search results list. Each cluster corresponds to a topic and contains one or more items appearing to the search results list. Thus, searchers are able to filter out results belonging to a specific topic. Such clusters derive from the short descriptions that accompany each item returned from the underlying search engines. However, due to the machine-generated nature of the clusters, it is difficult to provide semantically distinguished clusters with labels corresponding to the actual meaning of their content.

2.3.5 Query refinement

Another effort towards improving search engines' retrieval performance dictates the refinement of user queries with semantically related terms [11]. Most of the efforts in this direction concentrate on the disambiguation of the query terms based on either local (i.e., results sets) or global (usually ontologies expressed as thesauri) document analysis. Others proposed the utilization of lexical affinities to automatically refine queries [13], and the usage of both the text surrounding the query terms in the search results and the text surrounding the query term in the document being read [16]. Recently, there have been efforts that utilize ontologies for finding query related terms in order to improve retrieval efficiency [14,19]. However, when it comes to large-scale web search engines, the utilization of ontologies in query construction methods is difficult for three reasons [5]: (i) integration is extremely hard, (ii) the web imposes scalability and performance restrictions and (iii) there is a cultural divide between the semantic Web and information retrieval disciplines.

– Google's approach
  During the past few years, major large-scale web search engines and especially Google that seems to be the most popular one,[9] have evolved their provided functionality. Although the mechanics of their approaches are not officially published, it is evident that some of the above techniques (i.e., search personalization, relevance feedback, human intervention in ranking) are finding their way into the provided searching process.
  More specifically, Google's "+1" (successor of discontinued Stars and SearchWiki[10]) approach takes advantage of Google Account authentication services to identify the searchers and consequently log their personal search tactics. Such information is also available to each user as his search history.
  As far as the query construction phase of a search session is concerned, major web search engines have made considerable progress. Autosuggest functionality within the search box is currently provided by default. According to

[3] http://www.stumpedia.org
[4] http://www.anoox.com
[5] http://www.irazoo.com
[6] http://www.mysidekick.com
[7] http://search.wikia.com
[8] http://www.yippy.com

[9] www.alexa.com, accessed at: 22 March 2012
[10] http://www.google.com/psearch

Google's approach, upon issuing a query, a list of query suggestions is displayed. Consequently, users can rapidly express their initial query by selecting and promoting the suggestion that best suits their information needs. Moreover, Google provides the option to fetch search results while users type their query.

In this paper, we introduce a query construction service suitable for large-scale web search engines that is based on Linked Open Data – LOD [9]. The service utilizes datasets provided by DBpedia [4]. The service incorporates an interactive, easy to learn, non-intrusive GUI via which searchers obtain information about the semantics of their search terms as well as alternative wordings for verbalizing their search intentions.

3 Query construction service

In this section, we address the query formulation problem and introduce a service that assists searchers pick the most suitable terms for formalizing their requests regardless of the seeking mode to which they engage every time. To build our query construction service, we rely on the exploitation of collaborative knowledge recorded in Wikipedia and which is available via DBpedia's datasets. Nevertheless, our approach can be easily extended to incorporate other collaborative datasets such as YAGO [26]. Serving as an application of the semantic web, the proposed approach provides an interactive GUI that seamlessly integrates the knowledge provided by web users with large-scale web search engines. Such knowledge is modeled in a carefully designed, modular conceptual schema that supports querying against a large volume of linked data.

To develop our schema, we made use of N3-formated[11] datasets provided by DBpedia. The details of the schema construction are given in Sect. 3.1.

Searchers are able to express their information needs by interacting with an accordingly designed, LOD browsing GUI. More specifically, interactions through the GUI are converted to query/response pairs that are administered by a middleware. Queries are encapsulated in http-GET requests and responses are expressed as xml-based strings. The middleware addresses the queries to the underlying datastore, which, in turn, delivers the appropriate responses to the middleware. Such responses are converted to xml and channeled back to the GUI.

Finally, the GUI transforms responses to keyword-based queries and addresses them to the underlying large-scale web search engine (see Fig. 1). A detailed description of the proposed approach is provided in the following sections.

[11] N3 notation, Berners-Lee, T.: Notation 3 (N3): A Readable RDF Syntax. www.w3.org/DesignIssues/Notation3.html

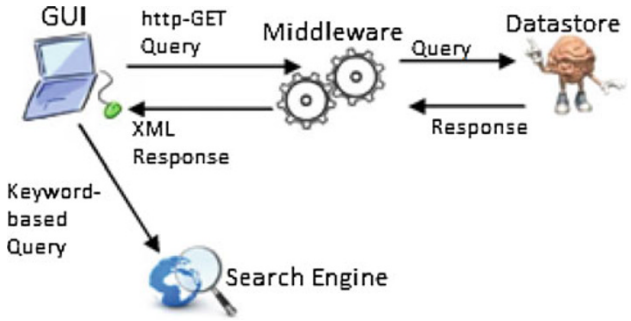

Fig. 1 Proposed service's architecture

Table 1 Statistics of the DBpedia datasets

Datasets	No. of items
Wikipedia articles	2,866,994
Disambiguation entries	226,978
Categories entries	339,112
WordNet classes	124
Articles linked to WordNet classes	497,797
Infobox records	19,230,789

3.1 A Wikipedia-based schema

As previously mentioned, the proposed system stores linked data originating from DBpedia into a datastore that is based on a conceptual schema. Several studies exist that rely on DBpedia datasets for building highly expressive ontologies via the combination of Wikipedia and WordNet.[12] Two of the most widely known resources that have emerged from such efforts are the Kylin Ontology Generator (KOG) [28] and the YAGO ontology [26]. Based on the success of the above studies, we decided to take advantage of the knowledge hidden within various DBpedia datasets in favor of a query construction service that acts as a mediator between searchers and large-scale web search engines.

So far, the service utilizes the following DBpedia datasets[13]: (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 (v) the articles' infoboxes that contain semantically rich properties about the considered articles. Table 1 summarizes the statistics of the DBpedia datasets that are employed in this work.

These datasets are organized in a conceptual schema as Fig. 2 illustrates. Thus, the classes of the schema are:

[12] http://wordnet.princeton.edu

[13] http://wiki.dbpedia.org/Downloads32

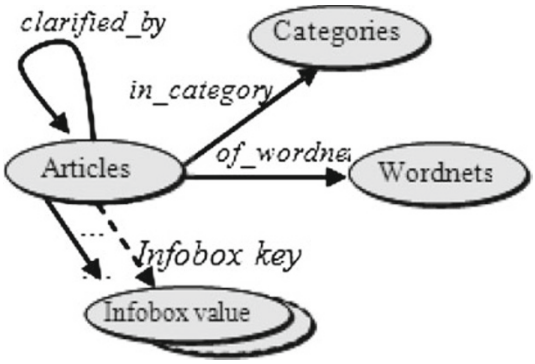

Fig. 2 Conceptual schema

(i) *Articles* that correspond to the Wikipedia articles organized as class instances, (ii) *Categories* that correspond to the appropriate categories of the Wikipedia articles, (iii) *WordNet* that stores the type of every Wikipedia article.

Moreover, the relations of the schema are: (i) "*clarified by*", a reflexive relation corresponding to the disambiguation pages of Wikipedia articles and which has the class 'Articles' as both domain and range, (ii) "*in category*" that connects every article to one or more appropriate categories. Another relation is (iii) "*of WordNet*" that connects articles associated with WordNet classes to an appropriate entity type.

Finally, the Wikipedia infobox properties are key–value pairs that are expressed as datatype properties of their corresponding article instances.

Based on the above schema, a datastore is created that is accordingly incorporated into the proposed query construction service in the hope of assisting searchers decipher the semantic orientations of their candidate queries before these are actually addressed to the search engine. The datastore is serialized as a MySQL database, taking in this way advantage of its fast indexing capabilities.

Given the highly dynamic nature of collaborative knowledge on the web, the proposed query construction service is designed in a way that it can be easily extended with existing and yet-to-appear datasets. Specifically, in case of an incoming dataset, a new class would be added to the schema (see Fig. 2) together with its corresponding relation. Moreover, a new query–response pair would be defined together with the corresponding rendering of the response from the client-side GUI, as will be explained later in this paper.

Next, we present the GUI of the proposed query construction service and illustrate through several examples how it contributes to the overall search process on the web.

3.2 GUI for structuring queries

So far, we have discussed the motive of our work and described the core semantic data that are employed to identify the semantics of the queries that are addressed to the underlying search engine. We now turn the discussion to the description of the proposed GUI, which extends the traditional input box of large-scale web search engines by (a) suggesting context-aware formulations of the intended queries based on the first letters that are inserted into the input box, and (b) visualizing the semantics of the initial query. The design principles require that the GUI should be interactive, inductive, easy to use and fast to execute. Having such requirements in mind, we initially built the auto-suggest input box illustrated in Fig. 3.

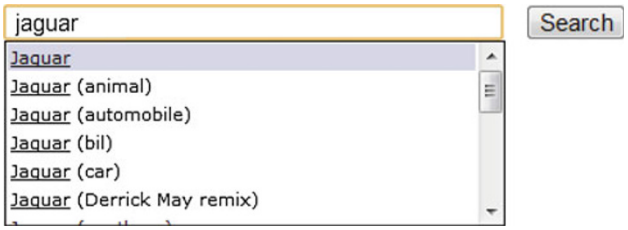

Fig. 3 Automatic suggestions for query construction

This box enables searchers to type their search terms and receive in response a set of alternative query wordings. In particular, upon typing a few characters of a search query, the box suggests a number of strings that can be attached to the typed tokens to complete them. The auto-complete suggestions are leveraged from the titles of the Wikipedia articles that the service encapsulates. In case the searcher does not wish to employ any of the suggested query alternatives, he can ignore the suggestions and search with his self-selected keywords.

Up to this point, the described functionality replaces the autosuggest functionality that has been recently added to major large-scale web search engines such as Google and Yahoo! with autosuggest functionality powered by Wikipedia.

If the searcher decides to select one of the offered suggestions, an http-GET query is issued to the middleware, which, in turn, replies with an xml-encoded response containing information that derives from the underlying datastore. The GUI visualizes the responses as interconnected boxes located above the search engine's input box. Each box has a title corresponding to an article from Wikipedia and a number of labels beneath it pertaining to the article's possible semantic relations (i.e., disambiguations, categories, infobox properties) to other elements. Searchers are able to interact with the boxes by clicking on a relation. In that case, the initial query is reformulated. As stated earlier in this paper, there are currently four different types of relations (i.e., disambiguations, categories, WordNet-classes and infoboxes) implemented, although the modularity of the proposed service allows for further expansion with more relations.

– Disambiguations/WordNet classes
  If the searcher clicks on a "clarified by" relation, query disambiguation is performed as follows: at first, the

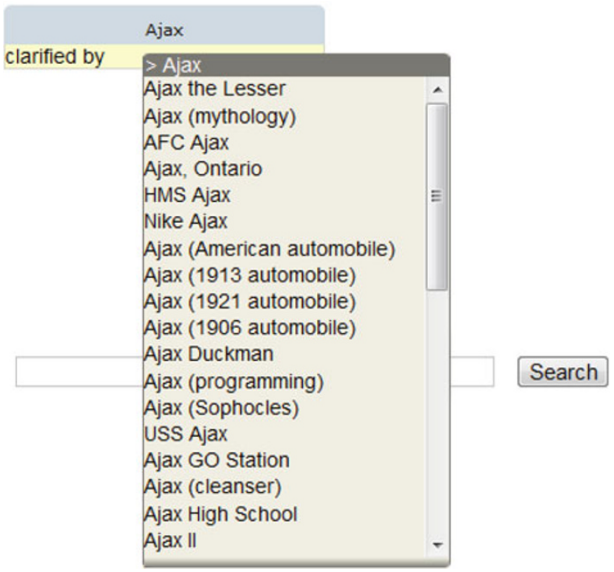

Fig. 4 Provided query disambiguations

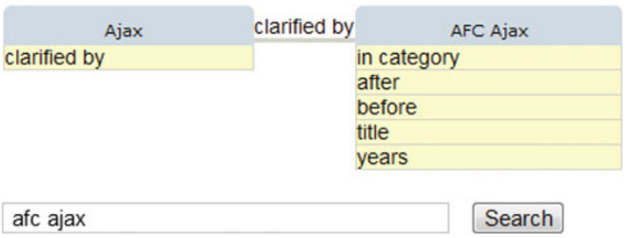

Fig. 5 Selected query disambiguations

searcher is presented with a list of all the corresponding disambiguations that match his selected suggestion (Fig. 4). Such disambiguations could be grouped by WordNet classes, provided they share 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 (Fig. 5), which is connected to the previous box with a line labeled "clarified 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 search engine.

– Categories
In a similar manner, if the searcher clicks on an "in category" relation, a pop-up menu appears containing the corresponding categories. Upon selecting one, a new box named after the selected category is sketched at the right, which is connected to the previous box with a line labeled "in category" (Fig. 6). Simultaneously, a search query consisting of keywords deriving from the two box titles

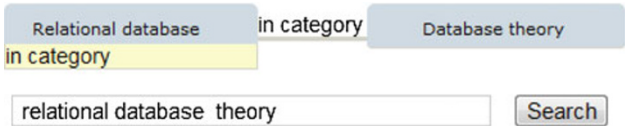

Fig. 6 Selected category

Fig. 7 Infobox properties

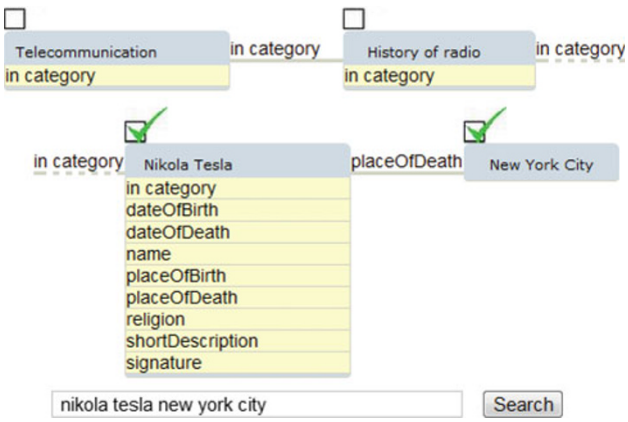

Fig. 8 Selecting query terms

(elimination of duplicates is applied) is addressed to the underlying search engine.

– Infoboxes
Finally, if the server's response consists of infobox properties realized as key–value pairs, the keys are displayed as labels. If the searcher clicks on a key, its corresponding value(s) appear(s) (Fig. 7). Upon selecting a value, a new box containing the selection is sketched, which is connected to the previous box with a line named after the infobox property's key.
Then, a search query consisting of the keywords deriving from the two box titles (elimination of duplicates is applied) is addressed to the underlying search engine.
Each sketched box corresponds via its title to a part of the resulting query. The searcher controls the participation of each box in the search query by clicking on the checkbox that resides on top of each box (Fig. 8).

This way, the searcher is provided with information for determining and accordingly expressing the semantic orientation of his queries, before/while these are issued for search.

Next, we present the middleware of the proposed query construction service and illustrate how it contributes to the overall search process.

3.3 The Middleware

As stated in the previous section, each interaction between searchers and the GUI results in an http-GET request that is addressed to the middleware. Depending on the request, the middleware issues appropriate queries to the underlying datastore. Since the datastore is serialized in MySQL, such queries are transformed to SQL-select statements. The results of each statement are encoded by the middleware in xml-based strings that are routed back to the GUI. Currently, there are four types of requests supported. These are outlined below together with their corresponding xml-encoded responses:

(a) article disambiguations
In case of a request for disambiguation, the http-GET string contains parameters determining (a) the id of the request, (b) the type of the request (i.e., disambiguation) and (c) the name of the Wikipedia article for which the disambiguations are requested. The corresponding response is an xml-based string containing the possible disambiguations as a set of <instance> elements. Below, the request: id = "q0", type = "disambiguation", name = "Ferrara" results in the following xml-string:

```
<reply type='success' iid='Ferrara'
 rid='disambiguation' id='q0'>
<instance  id='Ferrara'>
<label lang='en'>Ferrara</label> </instance>
<instance  id='Ferrara_Fire_Apparatus'>
<label lang='en'>Ferrara Fire Apparatus</label>
</instance></reply>
```

(b) article WordNets
In case of a request for WordNet categories, the http-GET string contains parameters determining (a) the id of the request, (b) the type of the request (i.e., WordNet category) and (c) the name of the Wikipedia article for which the WordNet categories are requested. The corresponding response is an xml-based string containing the possible WordNet categories as <instance> elements. Below, the request: id = "q0", type = "in_wordnet_category" name = "Ferrara" results in the following xml-string:

```
<reply type='success' iid='Ferrara'
 rid='in_wordnet_category' id='q0'>
<instance id='city'><label lang='en'>city
</label></instance>
<instance  id='monument'>
<label lang='en'>monument</label>
</instance></reply>
```

(c) article categories
In case of a request for categories, the http-GET string contains parameters determining (a) the id of the request, (b) the type of the request (i.e., category) and (c) the name of the Wikipedia article for which the categories are requested. The corresponding response is an xml-based string containing the possible categories as <instance> elements. Below, the request: id = "q1", type = "in_category" name = "amade camal" results in the following xml-string:

```
<reply type='success' iid='Amade_Camal'
 rid='in_category' id='q1'>
<instance id='1956_births'><label lang='en'>
1956 births</label></instance>
<instance id='Living_people'><label lang='en'>
Living people</label></instance>
<instance id='Muslim_activists'><label lang='en'>
Muslim activists</label> </instance>
<instance id='Mozambican_politicians'>
<label lang='en'>Mozambican politicians
</label></instance></reply>
```

(d) key–value pairs harvested from the contained infobox

Finally, in case of a request for infoboxes, the http-GET string contains parameters determining (a) the id of the request, (b) the type of the request (i.e., infobox) and (c) the name of the Wikipedia article for which the infoboxes are requested. The corresponding response is an xml-based string containing the possible key–value(s) pairs of the infobox as <dtprop> elements (The key of the infobox is the value of dtprop's attribute "id" and the value of the infobox is the value of the dtprop element). Below, the request: id = "q2", type = "infobox" name = "Zathras" results in the following xml-string:

```
<reply iid='Zathras' qtype='infobox' id='q2'
 type='success'>
<dtprop id='affiliation'>Great Machine </dtprop>
<dtprop id='finish'>War Without End</dtprop>
<dtprop id='name'>Zathras</dtprop>
<dtprop id='planet'>Unknown</dtprop>
<dtprop id='portrayer'>Tim Choate</dtprop>
<dtprop id='race'>Unknown</dtprop>
<dtprop id='start'>Babylon Squared </dtprop>
</reply>
```

4 Evaluation

Recently, many research efforts have emerged that try to take advantage of the knowledge contained within collaborative systems such as Wikipedia in favour of information retrieval. We provide a brief overview of some of the most interesting approaches, which is focused on the corresponding assessment method each approach has followed. In [19], the Koru search interface is introduced that offers WikiSauri, i.e., thesauri extracted from Wikipedia articles, upon which users rely to find alternative query formulations for satisfying their

search intentions. Queries are addressed towards an indexed corpus. The designers of Koru evaluated their system using the 2005 TREC HARD track [1]. More specifically, they performed a human study for which they calculated recall, precision and F-measure, averaged over all documents in the underlying index.

Similar assessment approaches have been followed in [17] and [30]. Both of these approaches employ Wikipedia to aid searchers construct/reformulate queries that are addressed to an underlying document repository. More specifically, in [17], Wikipedia is employed to strengthen weak ad hoc queries addressed to an index of Wikipedia articles. In [30], Wikipedia is employed to realize pseudo-relevance feedback by categorizing queries that are addressed towards an index of datasets provided by several TREC tracks.

Along these lines, Ester [6] provides a search interface for combined full-text and ontology search. More precisely, Ester takes as input a corpus and an ontology. The corpus is indexed and the YAGO ontology is employed to provide query construction functionality over the provided search engine. The designers of Ester assess the efficiency of the system and the quality of the search results, which, in turn, breaks down to: (a) the quality of the ontology, (b) the quality of the entity recognizer (i.e., the correct disambiguation of each query term) and (c) the quality of the combination of both the ontology and the full-text queries. According to the authors of [26], popularity settles for the quality of the employed ontology (i.e., YAGO). As far as the quality of the entity recognizer is concerned, the authors of [26] measured the precision of their approach. Finally, to assess the quality of the combined ontology and full-text queries, they calculated the recall of their system together with a metric called P@10, which is based on the assumption that the top ten entities mentioned on the respective Wikipedia page are considered relevant to the corresponding query term.

The common ground of the above approaches is that they all try to integrate an external corpus such as Wikipedia with an indexed repository to come up with a semantically rich search service. The repository's index is designed from scratch to meet the requirements of the corresponding search service. The assessment of the majority of the aforementioned approaches is based on the calculation of traditional IR metrics (such as precision and recall), which are accordingly compared against well-known evaluation tracks (e.g., TREC HARD track). Consequently, a large part of the assessment results refers to the quality of the index, instead of the quality of the search service per se.

The proposed approach, despite the fact that it follows a similar pattern (i.e., integration of Wikipedia-based information with a large-scale search engine on the web), cannot be assessed according to a similar scenario, due to the fact that there is no access available to the index of the underlying search engine. Consequently, metrics such as recall or F-measure cannot be calculated.

Having the above thoughts in mind, we discuss the details of an evaluation we carried out to validate the contribution of the service to the query construction process. The evaluation consists of a qualitative analysis of the autosuggest service and a human survey. Although a survey cannot provide definite conclusions about a service that is practically addressed to the entire web population, we believe that the findings give some indication about the usefulness of the proposed service in the query construction process.

4.1 Evaluation process

The query construction service is a two-step process. Initially, it provides autosuggest functionality by responding to the corresponding keystrokes of an information seeker. Prefix search is performed to an index that is composed of words and/or phrases originating from Wikipedia. Then, upon selection of a suggestion, the information seeker is offered the chance to modify the initial query through the appropriate interactions that are provided by the service.

Thus, the evaluation of the proposed service is made up of two parts. The first part consists of a qualitative analysis of the provided autosuggest functionality. The second part consists of a human survey that aims at estimating the overall opinion of experienced web searchers about the other interactions of the service.

4.1.1 Qualitative analysis of the proposed autosuggest functionality

Major web search engines provide query suggestions to speed up the process of query construction. The mechanics of their corresponding autosuggest services are not formally disclosed. However, according to various insights,[14] such suggestions most likely derive from some kind of statistical analysis of the queries that have been addressed towards the search engine over time. Thus, it seems that a typical autosuggest service performs well when the searcher picks query suggestions that correspond to:

1. popular queries in general (e.g., "beatles"),
2. popular queries within the geographic region of the client computer that invoked the search engine (e.g., "coupons uk" when the search is performed within the United Kingdom),
3. queries that have been addressed to the search engine before from the same user (i.e., personalization),

[14] How Google Instant's Autocomplete Suggestions Work, available at: http://searchengineland.com/how-Google-instant-autocomplete-suggestions-work-62592

4. queries that contain unambiguous words (like e.g., "afghanistan").[15]

The above criteria promote query suggestions that are affected mainly by popularity. However, it is argued that there are also times when popularity stands in the way between searchers and meaningful query suggestions.

Such an occasion happens when the searcher's information needs correspond to a word or phrase with various meanings (or "senses") and the searcher is interested in a less popular one. For example, an information need about "jaguar" the animal (not the car) corresponds to the suggestion "jaguar" from Google, which, in turn, corresponds to a search results list (at least within the scope of the first page) full of resources about the famous car and just one resource about the animal. Suggestions lead to even more useless search results as the number of possible meanings of an ambiguous word rises. The situation gets even worse when a popular resource (e.g., movie) is named after a word that literally means something else. Consider, for example, the term "ajax", which has more than 20 senses according to Wikipedia[16] (see Fig. 4).

The proposed approach introduces an autosuggest service that considers the semantic flavor of the query suggestions that are recommended to the searchers by matching their input against Wikipedia's titles. Ambiguous terms are properly disambiguated and the deriving disambiguations are prioritized within the list of the provided suggestions. Thus, the searcher's input "jaguar" leads to a list of disambiguated query suggestions that contain the not-so-popular information need "jaguar (animal)" (see Fig. 3), which, in turn, results in a list full of useful resources about such a need. The service is benefited from the fact that disambiguated words/phrases in Wikipedia appear as article titles that provide contextual words within parentheses after the ambiguous word/phrase. Thus, an autosuggest service that performs prefix search over an index of such literals results in a list of semantically disambiguated suggestions.

It is obvious that the proposed autosuggest service cannot handle the same amount of cases as compared against major web search engines, since the size of the underlying information spaces are hardly comparable. Moreover, in many cases, popularity-based factors succeed in predicting the right suggestions. However, it is evident that there is also a considerable number of cases where statistical analysis largely based on popularity is less effective than the approach followed in this paper.

[15] Apart from the above factors there may be other, statistical factors that affect the quality of query suggestions

[16] Ajax disambiguation page in Wikipedia: http://en.wikipedia.org/wiki/Ajax

The following section presents a human survey aimed at evaluating the other interactions of the proposed service with respect to Spencer's information-seeking modes.

4.1.2 Human survey

The primary goal of this survey is to assess the semantic enhancement of the query construction phase of a traditional search session, as provided by the proposed approach. Along these lines, a simple and fast to execute questionnaire was compiled, destined to be answered by web users coming from arbitrary backgrounds that share the common habit of spending considerable amount of time online. The questionnaire was promoted through social networks and mailing lists. Brief instructions about using the service were attached directly to the service's web site during the evaluation.

Although the service under evaluation is practically addressed to the entire web population, it was decided to disseminate the corresponding questionnaire just to experienced web searchers who are accustomed to the basic web metaphors that are employed by the proposed service. This way, it was anticipated that the participants would have no trouble in understanding the provided functionality and accordingly assess it. On the other hand, less experienced web searchers could provide more insights about the usability of the proposed service.

The evaluation process dictates that each participant should employ the proposed service as many times as required to satisfy a specific information need. The consecutive interactions of a web searcher with the proposed query construction service in the context of satisfying a specific information need is considered as a search session. When the search session is completed, the participant is asked to fill in the questionnaire to answer questions about the overall searching experience. Thus, each record within the survey results refers to a participant's opinion about the service for a specific search session corresponding to a single information need.

– Questionnaire

The questionnaire is designed so that it can be rapidly answered. It comprises two questions following the Likert scale, two yes/no questions and three closed questions. The questionnaire was completed 106 times.

The first question (i.e., Q.1 "Which of the following statements describes best your initial search intention?") is meant to rank the search sessions according to Spencer's information-seeking modes. The second question (i.e., Q.2 "At the end of the searching process, was your information need satisfied?") is meant to assess the overall satisfaction of the participant. The third question (i.e., Q.3 "Did you modify your initial query?") is meant to determine whether the

search session was concluded in a single step. In case of a negative response, the search session is concluded and the participant can either return to the service to engage himself in another session, or abandon the service. In case of a positive response, the participant is prompted to answer the rest of the questions. The fourth question (i.e., Q.4 "Where did you find the terms that modified your query?") is meant to determine the origin of the terms that refine the initial query. Thus, the participants can pick: (a) from the proposed service, (b) from the search results, (c) from an outside source, or d) from a combination of the above. If the refinement of the initial query is based on the proposed service (i.e., answer a), the participant is prompted to answer the next question (i.e., Q.5 "How were the terms supplied by the service?") to rank by popularity the interactions provided by the service. The next question (i.e., Q.6 "Did you click on any of the checkboxes on top of the sketched boxes?") is meant to determine whether the participants believe that they are always in control of the resulting query; they can decide which terms should participate in the final query by checking or unchecking the corresponding checkbox. Finally, the last question (i.e., Q.7 "Do you think that the provided service was easy to use?") refers to the usability of the proposed service.

– Results assessment

The 106 answers to question Q.1 are distributed to Spencer's information-seeking modes as presented in Table 2.

It is evident that most search sessions refer to the exploratory mode, followed by the known-item mode and the "Don't know what I need to know" mode. However, all three modes contain enough data for a proper evaluation, which is reviewed below.

According to the answers of question Q.2 (see Table 3), the overall impression of the participants about the service is rather positive.

The answers to question Q.3 (see Table 4) indicate that almost a quarter of the search sessions contained only the initial query.

It is interesting to observe that 66.67 % of the participants who concluded their search session in one step (i.e., answered "No" to question Q.3) engaged themselves to the "known-item" seeking mode. This is indicative of the fact that search engines seem to perform well when searchers know what they are searching for and how to express it in keywords. Moreover, it is believed that many participants chose not to complete the questionnaire instead of negatively answering question Q.3 in case they did not get a suitable suggestion from the autosuggest input box of the service. Thus, it is not safe to conclude that nearly three-quarters of the search sessions employed the proposed service.

The answers to question Q.4 (see Table 5) indicate that most of the reformulated queries employed terms derived from the interactions of the participants with the proposed service.

It is evident that exhaustively running through the search results to find terms that will refine a poorly chosen initial query is the last option for the participants of this survey. Moreover, the provided functionality seems to successfully address this drawback of current large-scale web search engines. In fact, the effectiveness of the service derives from the fact that most of the 54 participants who employed the proposed service during a search session (total responses of first answer in Table 5) satisfied their information needs. This conclusion derives from the restriction of answers to question Q.2 to the 54 search sessions that employed the service (see Table 6).

Additionally, the combination of answers to Q.4 and Q.1 draws interesting conclusions about the origin of the terms

Table 2 Distribution of search sessions according to Spencer's seeking modes

Seeking mode	Responses	(%)
Known item	34	32.08
Exploratory	47	44.33
Don't know what I need to know	25	23.58

Table 3 Answers to question Q.2 "At the end of the searching process, was your information need satisfied?"

Satisfied	Responses	(%)
Strong disagree	3	2.83
Disagree	11	10.38
Neutral	23	21.70
Agree	55	51.89
Strong agree	14	13.21

Table 4 Answers to question Q.3 "Did you modify your initial query?"

Answer	Responses	(%)
Yes	79	74.53
No	27	25.47

Table 5 Answers to question Q.4 "Where did you find the terms that modified your query?"

Answer	Responses	(%)
I picked from the ones provided by the service	54	72.00
I picked them from the search results	5	6.7
I picked them without any help from the service or the search results	8	10.67
Combination of the above	8	10.67

Table 6 Answers to question Q.2 restricted to the 54 search sessions that employed the proposed service

Satisfied	Responses	(%)
Strongly disagree	1	1.85
Disagree	6	11.11
Neutral	10	18.52
Agree	30	55.56
Strongly agree	7	12.96

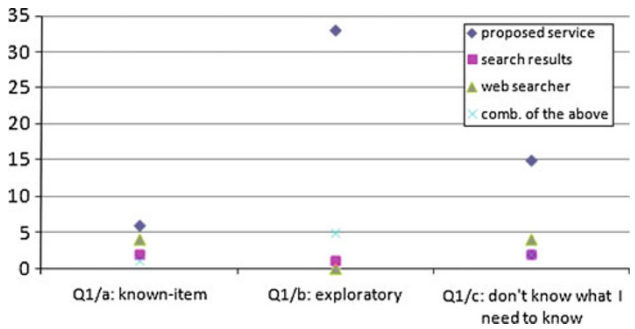

Fig. 9 Distribution of responses to Q.4 (i.e., origin of query refinement terms) across Spencer's seeking modes. Y-axis holds the number of participants, whereas X-axis holds the information-seeking modes

Table 7 Answers to question Q.5 "How were the terms supplied by the service?"

Through the	Responses	(%)
'clarified by' option	13	24.07
'in category' option	34	62.96
'infobox' option	2	3.7
Combination of the above	5	9.25

that have been employed to reformulate an initial query across Spencer's different seeking modes [25]. More specifically, as Fig. 9 shows, search sessions underpinning the exploratory mode are most likely to use terms derived from the proposed service and less likely to use terms derived from the search engine in order to refine the initial query. The same pattern (although in a lower scale) is evident in the "don't know what I need to know" information-seeking mode. Thus, it is apparent that search engines have a limited potential in improving an initially unsuccessful query. Such a problem is of minor importance for the 'known-item' seeking mode, where search engines seem to perform well.

The answers to question Q.5 (see Table 7) indicate that the most popular option for query refinement through the proposed service was the "in category" option, followed by the "clarified by" option. It seems that infoboxes did not provide significant aid to the participants.

The answers to question Q.6 (see Table 8) reveal that many participants did not necessarily terminate their search

Table 8 Answers to question Q.6 "Did you click any of the checkboxes on top of the sketched boxes?"

Answer	Responses	(%)
Yes	24	34.78
No	45	65.22

Table 9 Answers to question Q.7 "Do you think that the provided service was easy to use?"

Satisfied	Responses	(%)
Strongly disagree	0	0
Disagree	5	6.49
Neutral	12	15.58
Agree	43	55.84
Strongly agree	17	22.08

session by submitting the output of their final interaction. Instead, they chose to recall the output of one of their previous interactions during the same session. This underpins the fact that a highly interactive service like the one proposed in this paper should at all times allow the end user to be in control of the ongoing process.

Finally, answers to question Q.7 indicate that the overall impression about the usability of the service is rather positive (see Table 9). This is particularly important, since ease of use is one of the fundamental requirements of this service.

Although the number of participants in this survey cannot be compared against the actual number of web searchers who would employ this service in a real-world scenario, still a couple of insights derived from this survey seem beyond any doubt. First of all, large-scale web search engines seem incapable of aiding their users in case of a poorly articulated search query. Exhaustively running through the initial search results list in order to find more relevant terms discomforts searchers. Secondly, the proposed service seems to perform well in its quest for filling the semantic gap between the initial and the resulting query of a search session.

5 Discussion

The proposed work realizes a query construction service that is based on an external corpus (i.e., Wikipedia) to recommend semantically related terms to its users. The originality of the proposed approach as compared to similar works that have been presented in Sect. 2.3.5 lies in the fact that the external corpus is completely decoupled from the underlying document index. Such an approach successfully addresses the scalability issue mentioned in [5], since integration with large-scale web search engines is not affected by the size of the engine's index.

Moreover, the proposed GUI provides a way of traversing a semantic web structure (i.e., underlying schema) without revealing any of the specific terminologies that are quite commonly employed by similar approaches of the semantic web community, rendering in this way the service fast to learn and easy to use.

It should also be stated that by employing the proposed service, searchers are instantly acquainted with query terms that otherwise would take them a lot of time to gather by exhaustively running through the search results of potentially vague queries. Thus, the proposed service is particularly useful for the 'exploratory' and the "don't know what I need to know" information-seeking mode [16].

Additionally, the provided functionality is smoothly integrated into the traditional search engine's GUI, since it occupies just a small portion of the screen on top of the input box, thus leaving plenty of room for the search results.

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 to their prosperity.

Finally, it should be mentioned that if there is no available information about the user-typed terms, the overall search process does not break down and the query is transparently forwarded to the underlying search engine. Therefore, the worst case scenario is that searchers do not get any help from the service, but still their query is automatically submitted for search.

The proposed query construction service has been integrated so far with two major web search engines (Google and Yahoo!) and can be accessed online.[17] Thus, we believe that the integration is doable for any search engine that gives programmable access to the input box.

6 Conclusions

In this paper, a query construction service suitable for integration with large-scale web search engines is introduced in an attempt to semantically assist web information seekers in specifying precise and useful queries. This work is motivated by the fact that despite their wide appreciation, large-scale search engines entail practical limitations when employed by users experiencing certain search modes. The proposed service attempts to bridge the gap between large-scale web search engines and web searchers who are incapable of accurately verbalizing their information needs in effective search queries.

More specifically, the proposed query construction service relies on the semantic information provided by DBpedia and enables users to understand the semantic orientation of their search keywords before/while these are actually issued to the search engine. The service transparently delivers the provided functionality to web searchers through an interactive, non-intrusive and easy to use GUI. The proposed service was accordingly evaluated through a human survey consisting of 106 answered questionnaires. Although it is very difficult to perform a valid evaluation for a service that practically refers to the entire web population, the results from the evaluation underpin the motive of this work, i.e., the fact that large-scale web search engines encounter limitations when employed by users featuring specific search modes. Moreover, the overall outcome of the evaluation dictates that the proposed query construction service moves on the right tracks.

The proposed work points to directions for future work in a very important field of the web living at the intersection of large-scale search engines, collaborative knowledge and the semantic web. The prototype query construction service is only the first step towards this direction. Further steps are underway concerning the enrichment of the underlying datastore and the improvement of the provided interface both in terms of expressivity and ease of use.

[17] Demo, available at: http://thalassa.ionio.gr/snh/entry/

References

1. Allan J (2005) HARD Track overview in TREC 2005 high accuracy retrieval from documents. In: Proceedings of TREC-2005
2. Aktas M, Nacar M, Menczer F (2004) Using hyperlink features to personalize web search. Advances in Web Mining and Web Usage Analysis. In: Proceedings of the 6th International Workshop on Knowledge Discovery from theWeb, WebKDD 2004, Seattle
3. Anick PG (1994) Adapting a Full-text Information Retrieval System to Computer the Troubleshooting Domain. In: Proceedings of ACM SIGIR'94, pp 349–358
4. Auer S, Bizer C, Lehmann J, Kobilarov G, Cyganiak R, Ives Z (2007) DBpedia: a nucleus for a web of open data. In: Proceedings of the 6th International Semantic Web Conference
5. Baeza-Yates R, Ciaramita M, Mika P, Zaragoza H (2008) Towards semantic search, natural language and information systems. Springer Lect Notes Comput Sci 5039:4–11
6. Bast H, Chitea A, Suchanek FM, Weber I (2007) Ester: efficient search on text, entities, and relations, SIGIR'07, 671678
7. Bates M (1986) Subject access in online catalogs: a design model. J Am Soc Inform Sci 11:357–376
8. Belkin NJ (1980) Anomalous states of knowledge as the basis for information retrieval. C J Inf Sci 5:133143
9. Berners-Lee T, Chen Y, Chilton L, Connolly D, Dhanaraj R, Hollenbach J, Lerer A, Sheets D (2006) Tabulator: exploring and analyzing linked data on the semantic web. In: Proceedings of the 3rd Semantic Web User Interaction Workshop (SWUI) at ISWC
10. Bhogal J, Macfarlane A, Smith P (2007) A review of ontology based query expansion. Inform Process Manag 43(4):866–886
11. Billerbeck B, Scholer F, Williams HE, Zobel J (2003) Query expansion using associated queries. In: Proceedings of the ACM CIKM Conference

12. Bradley P (2008) Human-powered search engines: an overview and roundup, Ariadne, 54. http://www.ariadne.ac.uk/issue54/search-engines/. Accessed 22 March 2012
13. Carmel D, Farchi E, Petruschka Y, Soffer A (2002) Automatic query refinement using lexical affinities with maximal information gain. In: Proceedings of the 25th ACM SIGIR Conference, pp 283–290
14. Celik D, Elci A (2006) Discovering and scoring of semantic web services based on client requirement(s) through a semantic search agent. In: Proceedings of the 30th IEEE Computer Software and Applications Conference, pp 273–278
15. Daoud M, Tamine L, Boughanem M, Chebaro B (2009) A session based personalized search using an ontological user profile. In: ACM symposium on applied computing (SAC), Haiwai (USA). ACM Press, London, 10311035
16. Kawashige T, Oyama S, Ohsima H, Tanaka K (2006) Context matcher: improved web search using query term context in source document and in search results. In: Proceedings of the APWeb Conference, pp 486–497
17. Li Y, Luk RWP, Ho EKS, Chung FL (2007) Improving weak ad-hoc queries using wikipedia as external corpus. In: Proceedings of the SIGIR Conference, pp 797–798
18. Miller GA (1995) WordNet: a lexical database for English. Communications of the ACM 38(11):39–41
19. Milne DN, Witten IH, Nichols D (2007) A knowledge-based search engine powered by Wikipedia. In: Proceedings of the 16th Conference on Knowledge Management, pp 445–454
20. Papadakis I, Stefanidakis M (2008) Visualizing ontologies on the web, new directions in intelligent interactive multimedia, studies in computational intelligence, vol 142. Springer, Berlin, pp 303–311
21. Pazzani M, Muramatsu J, Billsus D (1996) Syskill and Webert: identifying interesting web sites. In: Proceedings of the 13th National Conference on, Artificial Intelligence, pp 54–61
22. Qui F, Cho J (2006) Automatic identification of user interest for personalized search. In: Proceedings of the 15th International World Wide Web Conference, pp 727–236
23. Salton G, Buckley C (1990) Improving retrieval performance by relevance feedback. J Am Soc Inform Sci 41(4):288–297
24. Sieg BM, Burke R (2007) Web search personalization with ontological user profiles. In: CIKM'07: Proceedings of the ACM Conference on information and knowledge management, New York, NY, USA, ACM, 525534
25. 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
26. Suchanek FM, Kasneci G, Weikum G (2007) YAGO: a core of semantic knowledge-unifying WordNet and Wikipedia. In: Proceedings of the WWW Conference, pp 697–706
27. Surowiecki J (2004) The Wisdom of Crowds. Random House, New York, ISBN 978-0385721707
28. Wu F, Weld DS (2008) Automatically refining the Wikipedia infobox ontology. In: Proceedings of the 17th International World Wide Web Conference, pp 635–644
29. Xu S, Jin T, Lau FCM (2009) A new visual search interface for web Browsing. In: Proceedings of the 2nd ACM International Conference on Web Search and Data Mining WSDM '09, pp 152–161
30. Xu Y, Jones GJ, Wang B (2009) Query dependent pseudorelevance feedback based on wikipedia. In: Proceedings of 32nd International ACM SIGIR'2009, pp 59–66