The Semantics of Query Modification

ABSTRACT

We present a method that exploits 'linked data' to determine semantic relations between consecutive user queries. Our method maps queries onto concepts in linked data and searches the linked data graph for direct or indirect relations between the concepts. By comparing relations between large numbers of user queries, we identify semantic modification patterns. The application of this method to the logs of an image search engine revealed interesting usage patterns, such as that users often search for two entities sharing a property (e.g., two players from the same team). These patterns can be used to generate query suggestions. Results of preliminary experiments show that the patterns enable us to generate suggestions for more queries than a method purely based on search-log statistics.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Query formulation*

Keywords

Query modification, linked data, query log analysis, query suggestion

1. INTRODUCTION

Users of search engines often need to engage in an iterative interactive process by providing a succession of queries so as to satisfy a single information need. The user submits a query, examines the retrieval results and then, depending on his or her satisfaction with the results, stops searching or modifies the query.

A number of studies have examined query modification behavior through the analysis of queries collected in search interaction logs (e.g., [17, 20, 21, 14, 15, 6]). In these studies query modifications are classified based on the overlap in terms between consecutive queries, by examining whether, compared to the previous query, terms have been added, eliminated, or substituted. Additions are interpreted as specifications, eliminations as generalizations, and substitutions as reformulations.

In this paper we propose a different approach that determines semantic relations between queries by exploiting the knowledge in a *linked data cloud* [3, 4]. In contrast to term-based methods, this allows us to determine also relations between queries that do not have terms in common but are semantically related, such as the queries `Andre Agassi` and `Boris Becker`. In addition, it allows us to recognize finer grained modification patterns than only addition, elimination and substitution. For example, our method identifies the relation between the queries `David Beckham` and `Victoria Beckham` not as a mere substitution, but as a relation between spouses. If this relation is observed frequently, we identify the semantic modification pattern that a query about a person is often followed by a query about this person's spouse.

A commonly used method to assist users with their query reformulations are query suggestions. When the user submits a query, the search engine provides (together with the retrieved results) a set of suggestions for follow-up queries [16]. In this paper we show that semantic query modification patterns can be applied to generate query suggestions. In contrast to existing suggestion methods that are purely based on statistics from query logs [16], semantic modification patterns enable us to also find suggestions for queries entered for the first time. Moreover, such patterns can generate suggestions that do not occur as queries in the log file, enhancing serendipitous discoveries. Finally, semantic patterns enable us to explain the relation between a query and the suggestions.

We continue this paper in Section 2 with a review of related work. In Section 3 we explain our method for extracting semantic query modification patterns and for using these patterns for query suggestion. In Section 4 we demonstrate the working of the pattern extraction method on log data of a large image provider and present results of preliminary experiments with query suggestion. Section 5 contains conclusions and discusses our results.

2. RELATED WORK

A number of studies have classified query modification patterns encountered in search logs of search engines [6, 8, 11, 14, 15, 17, 19, 20, 21]. These studies classify query modifications on the basis of terms in the queries. They examine whether queries have been added, eliminated or substituted compared to the user's previous query. When terms are added, the query is considered to become more specific (e.g., from query `Beckham` to query `Beckham Milan`), when terms are eliminated it is considered to become more general (e.g., from `Beckham Milan` to `Beckham`) and when terms are substituted a parallel movement is considered to be made (reformulation) (e.g., from `Beckham Milan` to `Beckham Madrid`). The large majority of the studies find that reformulations occur roughly twice as often as specifications which occur twice as often as generalizations [6, 8, 15, 17, 19, 20, 21].

Various other term-based modification patterns have been studied, including changes from singular to plural forms and vice versa, abbreviations, and spelling corrections (e.g. [14]). Like specification, generalization and reformulation, these modification patterns do not explain relations between queries that are similar at a semantic level but not at character level.

Although various authors have advocated the use of term-based modification patterns for enhancing query suggestion [14, 15, 20], so far this has not been realized. The most likely reason is that such query modification patterns are too generic to be applied directly: even if we know that most queries are followed by a substitution, we do not know which query terms need to be substituted by which new terms.

Existing techniques for query suggestion can be divided in document-based methods, log-based methods, ontology-based methods, and methods using combinations of these sources. Document-based methods consider the highest ranked documents returned by the search engine and suggest key phrases from these documents. The main drawback of this type of methods is that they cannot find suggestions that are conceptually related to the user's query, but do not co-occur in documents, as is often the case for synonyms [13].

Log-based query suggestion uses search statistics (e.g., [8, 13, 16]). In the simplest case, when the user enters query q, the queries that most frequently follow q are suggested. More advanced techniques include using random walks on graphs consisting of co-occurring queries [5] or consisting of queries and clicks on search results [1]. Log-based methods are adequate for queries that are entered frequently, as for these queries accurate statistics are available. However, for infrequent queries the statistics are based on only a few instances, which can lead to poor suggestions. For queries that have never been submitted before purely log-based methods cannot find any suggestions. This is a serious shortcoming, as research has shown that query distributions follow a power law, which means that a large proportion of the queries is unique [2].

Ontology-based suggestion techniques match the user's query to concepts in ontologies and suggest concepts that are linked to the matching concepts [12]. A disadvantage of this type of methods is that they are sensitive to the type of links in the ontology: if the ontology contains links that are not useful from the perspective of the user, the system generates irrelevant suggestions. For instance, DBpedia[1] contains a link from the movie '2001: A Space Odyssey' to 'United States' (being the country where the movie is created). Although the link is valid, the query `United States` is in most cases not a very good suggestion for users who search for the movie. Using relations that consist of multiple links leads to even stronger topic drift. To overcome these problems one needs to define the value of all types of links in the ontology.

The method presented in this paper combines the strengths of log-based and ontological query suggestion. The search statistics identify those semantic relations that have been relevant for users in previous sessions, while the semantics enable us to generate suggestions for first-time queries.

A key element of our method is the use of linked-data. Here we briefly review the main concepts of linked data; for a extensive overview we refer to [4]. The idea of linked data was first described by Tim Berners-Lee [3] in the form of four principles that prescribe how data should be published on the web. Following these principles ensures that the data can be easily shared with others, read by both humans and machines, and linked to data from other sources. Each entity in the data is referred to by a unique URI. Information about the entity can be attained by looking up the URI via HTTP. Information about entities and links between entities are coded in RDF [18]: a set of triples <subject, predicate, object>, where the subject and the predicate are both URIs and the object can either be a URI or a string literal. Examples of RDF triples are given in Figure 1. The first triple provides information about a single entity. The second triple provides a link between entities from different sources. The third triple states that two URIs from different sources refer to the same entity.

The Linked Open Data Project[2] aims to publish and connect as many open data sets as possible according to the linked data principles. Since the start of the project the number of data sets has grown explosively. The current size of the total data cloud is estimated at 4.7 billion triples [4]. The two largest data sets that we use in the experiments in this paper, DBpedia[1] and WordNet[3], are taken from this cloud.

3. METHOD

To determine how users of a search engine modify their queries, we extract the queries of individual users from a search log file, as we will explain in Section 3.1. Then, in Section 3.2, we map the queries on concepts in a linked data cloud and search the linked data to determine the semantic relation between pairs of consecutive queries. Finally, in Section 3.3 we count how often each type of relation occurs and determine patterns of semantic relations. In Section 3.4 we explain how the identified patterns can be applied for query suggestion.

3.1 Preprocessing

Before the actual analysis can take place the server logs of the search engine that is analyzed must be preprocessed. Cooley et al. [9] have described the various preprocessing steps in depth. Here we give a brief summary of the elements that are relevant for our purposes.

Server logs of search engines typically contain an entry for each query that is submitted through the engine and for each click that a user has made on a search result. Among other things, a log entry consists of the user's IP address, information about the browser (agent) that was used, the time of the request, and the submitted query or clicked result. Sometimes additional information about users is available,

[1] http://dbpedia.org/

[2] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[3] http://www.w3.org/2006/03/wn/wn20/

Subject:	`http://www.w3.org/2006/03/wn/wn20/instances/synset-soccer_player-noun-1`
Predicate:	`http://www.w3.org/2006/03/wn/wn20/schema/senseLabel`
Object:	`'soccer player'`
Subject:	`http://dbpedia.org/resource/Edwin_van_der_Sar`
Predicate:	`http://dbpedia.org/property/wordnet_type`
Object:	`http://www.w3.org/2006/03/wn/wn20/instances/synset-soccer_player-noun-1`
Subject:	`http://e-culture.multimedian.nl/ns/rijksmuseum/people5706`
Predicate:	`http://www.w3.org/2004/02/skos/core#exactMatch`
Object:	`http://e-culture.multimedian.nl/ns/getty/ulan#500011051`

Figure 1: Examples of RDF triples

coming, for instance, from browser cookies or log-in mechanisms.

We assign the log entries to user sessions. If cookies or log-ins are available users can be identified with certainty. Otherwise, we assume that all entries with the same IP address and agent are from one user. A new user session starts when there is a period of inactivity in the session longer than some predefined time interval. Finally, we list for each session the queries that are entered and conflate consecutive identical queries into one query.

3.2 Semantic relations between query pairs

The queries in the user sessions are mapped on concepts in a linked data cloud. Finding relevant concepts for queries is far from trivial, as it is often not clear what a user is exactly looking for when entering a query. We use the `rdfs:label` property of the concepts in the linked data to match the queries, as this property is meant to provide a human readable description for the concepts [7]. Queries are mapped on concepts that have an `rdfs:label` that is identical to the query. If no exact match can be found, queries are mapped onto concepts with labels that contain all query terms (after stemming). With this method each query is mapped onto zero, one, or multiple concepts. We purposely chose a conservative mapping method, sacrificing recall for precision, to reduce the amount of noise in the resulting modification patterns.

For each pair of queries that are consecutively submitted by the same user, we determine the semantic relation between the queries, as illustrated in Figure 2. A graph search algorithm is used for traversing the links in the linked data to find the shortest series of links that connects the two queries (their relation). As linked data graphs are often very large, measures have to be taken to keep the search tractable. We set the maximum number of links in a relation at 4. Pilot experiments showed that longer relations are hardly ever relevant. Equivalence relations, such as `skos:exactMatch` and `owl:sameAs`, indicate that two URIs refer to the same entity. The path search algorithm treats such equivalent entities as one.

Often multiple relations of the same length are found between two queries. For instance, two persons can both be of type soccer player and also both play in the same national team. All relations that are found for a pair of queries are taken into account, but in the rest of the analysis each relation receives a weight that is inversely proportional to the number of relations that are found for the query pair.

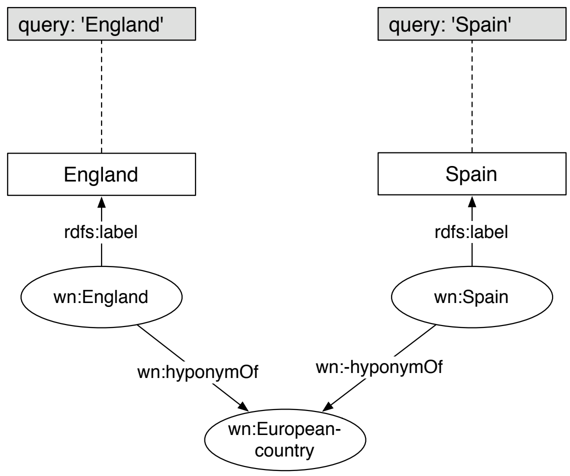

Figure 2: Example of a semantic relation between queries: the relation between queries `England` and `Spain` is that they both match hyponyms of the WordNet concept `European country`.

3.3 Patterns of semantic relations

The next step is to abstract away from relations between specific instances and infer *modification patterns* by removing the instances and keeping just the links. For instance, we may find that the relation from query `David Beckham` to query `Joe Cole` is that both refer to players in the English national football team:

```
David Beckham –DBpedia:nationalteam→
England_national_football_team
←DBpedia:nationalteam– Joe Cole
```

The arrows denote the directions of the predicates. This relation is abstracted to the modification pattern:

```
Q1 –DBpedia:nationalteam→ X
←DBpedia:nationalteam– Q2
```

To determine the importance of the patterns that are found, we count how often each pattern occurs between queries in the search log. The *support* of a pattern is defined as its relative frequency. The support value of each pattern is compared to a baseline that represents the expected frequency of the pattern. The baselines are computed by randomly sampling pairs of queries from different sessions in the log file and determining the relations between these pairs. We define the *confidence* of a pattern as the proportion of all (inter- and intra-session) query pairs matching the pattern that come from the same search session. Thus, if a pattern occurs in 3% of the query pairs where both queries come from the same session and 0.6% of the query pairs where the two queries come from different sessions, its support is 0.03 and its confidence is 0.03/(0.03+0.006)=0.83.

Finally, we apply an iterative process to improve the accuracy of the relations that we found. Patterns with high support but low confidence occur equally often between pairs from the same session as between pairs from different sessions, and thus are with high probability irrelevant patterns. We look up all query pairs for which relations are found that match an irrelevant pattern. We discard these relations and search the graph for other (longer) relations between the queries. When the new relations are determined, we recompute support and confidence. This process continues until the support and confidence of all patterns are above given thresholds or until no more relations are found. Finally, we output all patterns that are likely to be highly relevant, i.e. the patterns having both high support and confidence.

3.4 Applying semantic query modification patterns for query suggestion

To generate query suggestions we select the query modification patterns with support and confidence values above certain thresholds. When the user enters query q the query is mapped onto a set of concepts C in the linked data cloud as described in Section 3.2. The pattern with the highest support is applied to each concept c in C. For instance, the pattern `Q1`–`DBpedia:spouse`→`Q2` applied to c yields all concepts c' that are related to c via the relation `DBpedia:spouse`. Finally, the `rdfs:label`s of the concepts c' become the suggestions.

Often patterns do not result in suggestions for a query, for instance because the type of the query does not match the type of the pattern (e.g. when the spouse pattern is applied to concepts that are not persons) or because the relation is not defined for the particular concept (e.g. when the spouse pattern is applied to an unmarried person). In case the first pattern results in less suggestions than the number of suggestions that are allowed per query, the pattern with the second highest support value is applied. We continue to apply lower ranked patterns until we collected as many suggestions as allowed or until no more patterns are available.

The pattern-based suggestions that we find are primarily ranked by the support value of the pattern that generated them. When multiple suggestions are generated by the same pattern, they are ranked by the frequency of their occurrence in the log file: suggestions that occur most often as queries in the log file are placed at the highest rank. We combine the pattern-based suggestion method with a log-based method that suggests the queries which occur in the logs most frequently in the same session as the current query. The patterns are applied whenever the log-based method does not yield enough suggestions.

4. EVALUATION

We applied our semantic query modification method to the search log files of the commercial picture portal of a European news agency. The portal provides access to more than 2 million photographic images covering a broad domain. The log files record the search interactions of professional users (mainly journalists) accessing the picture portal. We used one year of search logs, containing 1,105,766 queries in 332,809 sessions. Search sessions were identified using a log-in and a browser cookie. A time-out of 15 minutes was used to determine session boundaries. The linked data consisted of various interlinked sources: the DBpedia Ontology[1], WordNet[3], the Cornetto Lexical Knowledge Base[4], the Getty[5] Thesaurus of Geographical Names, and the Getty Art and Architecture Thesaurus (AAT). Together these collections comprise 22 million RDF triples.

For the iterative process (Section 3.3) the support threshold was set at 0.0005 and the confidence threshold at 0.66667.

4.1 Query modification patterns

Table 2 (upper block) shows that for 55% of the 482,400 query pairs, both queries could be mapped onto concepts in the linked data cloud. For 45% of the query pairs for which concepts were found, a relation was found.

We manually evaluated the concepts found for 100 random queries (Table 2 middle block). We manually identified that for 74% of the queries a matching concept was present in our linked data. Our mapping method found a concept for 72% of the queries. For 89% of these queries at least one correct concept was found (precision). For some queries multiple concepts were found. When a correct concept was found, on average 85% of the concepts found were correct. Recall was 86%, meaning that in 86% of the cases in which a correct concept was in the linked data, it was also found. These results suggest that our mapping method is quite accurate, despite its simplicity.

We also evaluated the relations found for 100 random query pairs (Table 2 lower block). Our method found a relation for 39% of the query pairs and for 51% of these pairs at least one correct relation was found (precision). For pairs where a correct relation was found, the majority of the relations that were found were correct (74%). Recall was 63%. Although finding relations proved more difficult than finding concepts, we believe our method is accurate enough to find reliable query modification patterns. The incorrect relations

[4]http://www2.let.vu.nl/oz/cltl/cornetto/
[5]http://www.getty.edu

482,400 query pairs (all)	concepts found for both queries	0.55
	relation found	0.25
100 queries (manually assessed)	concept exists in linked data	0.74
	concept found	0.72
	precision	0.89
	recall	0.86
100 query pairs (manually assessed)	relation exists in linked data	0.32
	relation found	0.39
	precision	0.51
	recall	0.63

Table 2: Proportion of queries and query pairs for which concepts and relations are found in the linked data and the precision and recall of the found concepts and relations.

Table 1: Semantic modification patterns with the highest support.

	support	confidence	pattern
1.	0.031	0.94	`[ ]`
2.	0.017	0.99	`Q1 –DBpedia:spouse→ Q2`
3.	0.017	0.99	`Q1 ←AAT:distinguished_from- Q2`
4.	0.017	0.85	`Q1 –DBpedia:birthplace→ X ←DBpedia:birthplace– Q2`
5.	0.014	0.91	`Q1 –rdf:type→ X ←rdf:type– Q2`
6.	0.012	0.95	`Q1 –DBpedia:nationalteam→ X ←DBpedia:nationalteam– Q2`
7.	0.009	0.99	`Q1 –DBpedia:partner→ Q2`
8.	0.009	0.90	`Q1 –DBpedia:wordnet_type→ X ←DBpedia:wordnet_type– Q2`
9.	0.009	0.96	`Q1 –AAT:distinguished_from→ Q2`
10.	0.008	0.96	`Q1 –wordnet:memberMeronymOf→ X ←wordnet:memberMeronymOf– Q2`

appear to be of many different types, resulting in many incorrect modification patterns with very low support. Consequently, patterns that occur in high numbers are in majority based on correct paths.

The ten patterns with the highest support are shown in Table 1. The most common pattern was the empty relation ([ ]): two different queries that are mapped to the same concept, usually variant names for the same entity, such as 'Gent' and 'Gand'. Patterns 2 and 7 indicate that many users searched first on the name of a person and then on the name of his or her spouse or partner. Pattern 6 tells us that many users searched on two people from the same national team. Patterns 5 and 8 mean that users often search on two entities of the same type (e.g. both country names or both tennis players). Patterns 3 and 8 consist of the `AAT:distinguished_from` relation from the Getty Art and Architecture Thesaurus which links closely related terms, such as `prince` and `princess`.

22% of the identified relations are 'sibling relations': relations of the form `Q1–R→X←R–Q2`. Examples include patterns 4, 5, 6, 8 and 10 in Table 1. This shows that many people search for two entities with some common property, such as actors starring in the same movie. Another frequently occurring relation type (14%) are direct one-to-one or few-to-few relations, such as 'partner-of'. These types of modifications are apparently important for users, but cannot be identified with a term-based analysis of query modifications. Another observation is that most of the relations (74%) are complex, consisting of more than one link.

For validation, we applied our approach also to the logs of the Rijksmuseum web site[6] (an art museum), where similar types of semantic patterns were found.

4.2 Query suggestion

We used the identified query modification patterns for query suggestion. The support threshold of the patterns was set at 0.00085 and the confidence threshold at 0.85. This resulted in 44 modification patterns that were used to generate suggestions.

The log file was split in two parts. 80% of the search sessions were used as training data and the remaining 20% (64,767 query pairs) as test data. Semantic modification patterns and the co-occurrence statistics were extracted from the training set. These data were used to generate maximal 10 suggestions for each query in the test set. The suggestions are evaluated by counting the number of times at least one suggestion is found for a query (coverage) and the number of times the query immediately following the user's query in the session is among the suggestions (success rate). As baseline we used a purely co-occurrence-based method, which suggests the 10 queries that occur in the training data most frequently in the same session as the current query.

The success rate of the co-occurrence-based method was 0.33 with a coverage of 0.79. The pattern-based algorithm improved the success rate by 0.45%, though not significantly ($p = 0.22$). Coverage was significantly improved by 4.2% ($p<0.001$). When we only look at 'difficult queries', queries that occurred 5 times or less in the training data (36% of the queries), the success rate of the baseline algorithm drops to 0.093 with a coverage of 0.43. For these cases, the pattern-based algorithm significantly improved both the success rate (4.2%, $p=0.02$) and coverage (21.4%, $p<0.001$).

5. CONCLUSION AND DISCUSSION

We have shown the potential of combining statistical information gathered from search logs with semantic information. Using linked data for query log analysis enabled us to find patterns in query modification behavior that are interesting, non-trivial and easy to interpret. In contrast to traditional term-based approaches, this approach identifies relations between queries without any terms in common. This is a large advantage as our log data indicate that users often search for entities that are related semantically, but do not have common terms.

Initial results of applying semantic modification patterns to query suggestion demonstrate their practical value. Compared to purely statistical query suggestion, the patterns generate suggestions for more queries, improving success rate for infrequent queries and coverage for all queries.

We are currently conducting a more thorough study of the use of semantic modification patterns for query suggestion. We are investigating more advanced methods to combine the patterns with co-occurrence statistics. We are also studying the effects of the suggestion method in terms of diversity and serendipity. Another potential asset of using semantic patterns is their ability to explain the relationship between a user's query and the suggested queries. In other domains providing explanations has shown to enhance the users' acceptance of system-generated recommendations [10]. Through user experiments we will research the effects of explanations for query suggestions. Finally, all results will be validated on a second data set.

In this work we only studied the behavior of the user community as a whole. In the future we will extend this to query modification patterns that are specific for certain groups of users. Such patterns will potentially enable us to better understand differences between user groups and to generate personalized query suggestions.

[6]http://www.rijksmuseum.nl/

6. REFERENCES

[1] I. Antonellis, H. Garcia-Molina, and C.-C. Chang. Simrank++: Query rewriting through link analysis of the click graph. In *Proceedings of the 34th International Conference on Very Large Data Bases, Auckland, New Zealand*, pages 408–421, 2008.

[2] R. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA*, pages 76–85, 2007.

[3] T. Berners-Lee. Linked data: Design issues. http://www.w3.org/DesignIssues/LinkedData.html, 2006. last accessed November 5, 2009.

[4] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems, Special Issue on Linked Data*, in press.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 Workshop on Web Search Click Data, Barcelona, Spain*, pages 56–63, 2009.

[6] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. From 'dango' to 'japanese cakes': Query reformulation models and patterns. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence, Milan, Italy*, pages 183–190, 2009.

[7] D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF schema. http://www.w3.org/TR/rdf-schema/, 2004. last accessed November 5, 2009.

[8] P. Bruza and S. Dennis. Query reformulation on the internet: Empirical data and the hyperindex search engine. In *Proceedings of the RIAO'97 Conference on Computer-Assisted Searching on the Internet, Montreal, Canada*, pages 488–499, 1997.

[9] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1:5–32, 1999.

[10] H. Cramer, V. Evers, S. Ramlal, M. Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18(5):455–496, 2008.

[11] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38(5):727–742, 2002.

[12] L. Hollink, G. Schreiber, and B. Wielinga. Patterns of semantic relations to improve image content search. *Journal of Web Semantics*, 5(3):195–203, 2007.

[13] C.-K. Huang, L.-F. Chien, and Y.-J. Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, 54(7):638–649, 2003.

[14] J. Huang and E. N. Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China*, pages 77–86, 2009.

[15] B. J. Jansen, D. L. Booth, and A. Spink. Patterns of query reformulation during web searching. *Journal of the Americal Society for Information Science and Technology*, 60(7):1358–1371, 2009.

[16] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland*, pages 387–396, 2006.

[17] C. Jörgensen and P. Jörgensen. Image querying by image professionals. *Journal of the Americal Society for Information Science and Technology*, 56(12):1346–1359, 2005.

[18] G. Klyne and J. J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. http://www.w3.org/TR/rdf-concepts/, 2004. last accessed November 5, 2009.

[19] T. Lau and E. Horvitz. Patterns of search: analyzing and modeling web query refinement. In *Proceedings of the Seventh International Conference on User Modeling, Banff, Canada*, pages 119–128, 1999.

[20] S. Y. Rieh and H. Xie. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing and Management*, 42(3):751–768, 2006.

[21] M. Whittle, B. Eaglestone, N. Ford, V. J. Gillet, and A. Madden. Data mining of search engine logs. *Journal of the American Society for Information Science and Technology*, 58(14):2382–2400, 2007.