Semantic vs term-based query modification analysis

Vera Hollink
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
V.Hollink@cwi.nl

Theodora Tsikrika
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
Theodora.Tsikrika@cwi.nl

Arjen de Vries
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
Arjen.de.Vries@cwi.nl

ABSTRACT

Previous research has studied query modifications on a syntactic level by focusing on the addition, elimination and substitution of terms between consecutive queries that have at least one term in common. In this paper, we determine semantic relations between queries by first mapping them onto concepts in linked data sources and then identifying the relations between the concepts. This enables us to find relations between queries that do not share any terms. Moreover, with this approach we can find more detailed and more meaningful query modification patterns than with a term-based analysis. Application of our method to search logs of two search engines shows the importance of studying query modifications on a semantic level. Our results indicate that users often search for entities that are related semantically, but not syntactically. Specifically, users often successively search for two entities sharing a common property, such as two actors starring in the same movie, or two entities with a specific relation, such as spouses. We discuss the implications of these findings for the design of search engines.

Categories and Subject Descriptors

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

Keywords

Query modification, linked data, query log analysis

1. INTRODUCTION

Users of search engines often engage in an iterative interactive process by providing a succession of queries so as to satisfy a single information need. This search process is typically structured as follows: the user formulates and submits a query, examines the retrieval results, and then, depending on his (or her) satisfaction with the results, decides to either stop or to enter a new search cycle by modifying the query and re-submitting it in an attempt to reach a better outcome. Given that query modification is a key user behavior [10, 3, 14, 11], retrieval systems should provide assistance to their users during this challenging task. The study of query modification patterns allows us to gain insights into this user behavior, which can be used to improve the support that search engines offer to their users.

Query modification patterns are usually identified through the analysis of queries collected in search interaction logs [9]. Previous studies (e.g., [10, 14, 11]) have classified query modifications based on the overlap between terms in consecutive queries by examining whether terms have been added, eliminated, or substituted in a query, and interpreting additions as specifications, eliminations as generalizations, and substitutions as reformulations. The major limitation of such term-based methods is that they can only classify pairs of queries that have at least one term in common and, therefore, cannot determine the relations between queries that are semantically related without sharing any terms, such as the queries `Wim Kok` and `Ruud Lubbers`. Furthermore, such methods do not typically make any finer distinctions within each of the three main classes, even though there exist subclasses that correspond to very different user intents. For example, the modification of query `Posthuma Tour France` to `Posthuma Tour 2008` most likely indicates a second attempt to find information about the same event, while the modification of `Candy Dulfer` to `Hans Dulfer` signifies a shift of attention to another person. Nevertheless, term-based methods classify both cases as reformulations.

The work presented in this paper is also concerned with the identification of query modification patterns through the analysis of search logs. However, it goes beyond the term-based methods and proposes an approach that determines semantic relations between queries by exploiting the knowledge in a *linked data cloud* [1, 2]. Combining the semantic relations with statistics from the search logs allows us to recognize fine-grained and meaningful query modification patterns that are not visible from the usage statistics alone. For instance, the use of DBpedia[1] allows us to detect that many users successively search for two entities that have some property in common, such as both being soccer players in the same national team. In this paper we present our method for detecting semantic modification patterns and discuss the implication of such patterns for the design of search engines.

The remainder of this paper is structured as follows. In

[1] http://dbpedia.org/

Section 2 we present related work on query modification analysis and briefly explain the key ideas behind linked data. Section 3 presents our approach. In Section 4 we present results of applying our approach to the search logs of two search engines. In addition, we compare our findings to the results of a term-based analysis. The last section contains conclusions and discusses our results.

2. RELATED WORK

A number of studies have classified query modification patterns encountered in search logs of various types of search engines. Probably the most studied search logs are those of the Excite search engine [14, 15, 13]. Other general purpose engines that have been analyzed include Dogpile [10] and Yahoo! UK [3] and Yahoo! US [3]. Other researchers have examined the logs of search engines for limited domains, such as intranets [7] or commercial image providers [11]. Finally, Huang et al. [8] analyze logs of a group of users accessing a variety of search engines via a proxy.

Query modifications can be classified either manually [5, 13, 11, 14] or automatically [7, 15, 10, 3]. Manual classification is necessarily limited to a small number of queries, ranging from 2109 queries in [14] to 4690 queries in [13]. Automatic methods enable the analysis of much larger samples, up to 16 million queries in [3].

Studies that employ automatic methods usually classify query modifications solely on the basis of terms in the queries. These studies (as well as some of the manual studies) examine whether queries have been added, eliminated or substituted compared to the user's previous query [10, 15, 11, 7, 5]. When terms are added, the search is considered to become more specific (e.g., from query `Beatrix` to query `Beatrix holiday`), when terms are eliminated the search becomes more general (from `Beatrix holiday` to `Beatrix`) and when terms are substituted a parallel movement is made (reformulation) (from `Beatrix 2008` to `Beatrix 2009`). Some of the manual studies do not only look at terms but also classify modifications based on the meaning of the queries [14, 11, 13]. In these studies the same main classes are used, but a semantic modification from `Dog` to `Labrador` is also classified as a specification. An interesting intermediate approach is presented in [3]: this work aims at a semantic classification of query modifications into specification, generalization and reformulation by looking at the overlap in query terms, time intervals between queries and features of the user session as a whole.

The large majority of the studies find that the most frequently used modification type is reformulation, followed by specification and generalization [5, 13, 11, 14, 15, 10, 3]. Reformulations occur roughly twice as often as specifications which occur twice as often as generalizations. This finding is also supported by the work in [8], where it is observed that queries in the beginning of sessions tend to be more general than later queries. A noteworthy observation is that there is no difference in this respect between manual and automatic methods or between purely term-based and semantic methods. The only study in which different proportions are found is [7]: they find an equal number of reformulations and specifications and a much smaller number of generalizations. No explanation is given for this deviation.

Two studies have looked at the number of times users enter term variations [14, 5]. Such variations include, for instance, modifications from singular to plural forms or vice versa. Term variations are less common than the main modification classes, but they still make up a significant proportion of the queries, occurring about half as frequently as generalizations.

Sequences of query modifications are examined in [15, 10, 3]. Whittle et al. [15] found that users tend to repeat the same modification type (e.g., a specification is often followed by another specification). This is not confirmed by [3], who found that specifications are usually followed by generalizations and generalizations by specifications. This is also the dominant pattern in [10].

Lau and Horvitz [13] examined the relation between modification types and the time interval between submitting two consecutive queries. They find that specification is most likely after an interval of 20 to 30 seconds while reformulation peaks when the interval is longer than 5 minutes. However, the differences are quite small.

In summary, the four term-based modification patterns (specification, generalization, reformulation and term variation) have been extensively studied. Different authors have researched different variants and aspects of these patterns, but to our knowledge there are no papers that classify query modifications on a semantic level.

The key element that sets our approach apart from existing approaches and enables it to find semantic query modification classes is the use of linked-data. Below we will briefly review the main concepts of linked data. For a extensive overview we refer to [2].

The idea of linked data was first described by Tim Berners-Lee [1] 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 [12]: 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 and says that the concept `synset-soccer_player-noun-1` is described by the label `'soccer player'`. The second triple provides a link between entities from different sources, stating that Edwin van der Sar has the type soccer player in WordNet[2]. The third triple states that two URIs from different sources refer to the same entity.

The Linked Open Data Project[3] 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

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

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

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

total data cloud is estimated at 4.7 billion triples [2]. The two largest data sets that we use in the experiments in this paper, DBpedia[1] and WordNet[2], 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. 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 submitted by the same user. Finally, we count how often each type of relation occurs. In the rest of this section, these steps are explained in more detail.

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. [6] 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 that was used (called the *agent*), the time of the request and the submitted query or clicked result. Sometimes additional information about users is available, coming, for instance, from browser cookies or log-in mechanisms.

We group the log entries per user. 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 (typically 15 or 30 minutes). Finally, we list for each session the queries that are entered and conflate consecutive identical queries into one query.

3.2 Finding semantic relations between queries

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. In this study 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 [4]. We map queries on concepts that have an `rdfs:label` that exactly matches 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. Furthermore, we remove all concept-predicate pairs from the linked data that were present in more than 10,000 triples, as these relations are usually overly generic (e.g., stating that a person's gender is male).

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 (they are *smushed*). In other words, the equivalence relations are not reported in the relation between the queries and are not counted in the number of links that is followed.

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.

3.3 From relations to modification patterns

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:

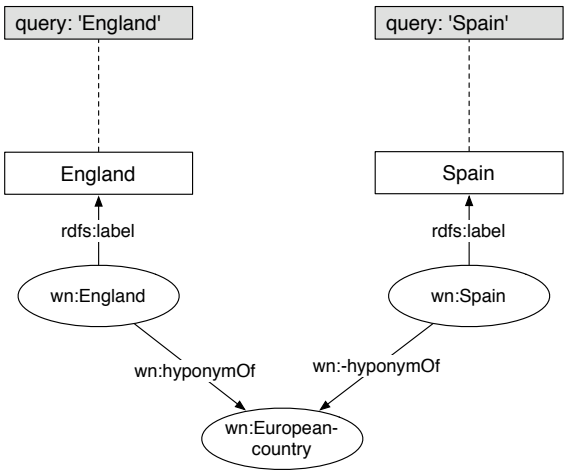

Figure 2: Example application of our procedure: the relation between queries `England` and `Spain` is that they both match hyponyms of the WordNet concept `European country`.

```
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. In this count all sessions are weighted equally, so that sessions with more queries do not contribute more to the final counts than shorter sessions. 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.

4. CASE STUDIES

4.1 Data sets

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. The linked data consisted of various interlinked sources: the DBpedia Ontology[1], WordNet[2], the Cornetto Lexical Knowledge Base[4] (which contains both Dutch and English terms), the Getty[5] Thesaurus of Geographical Names, and the Getty Art and Architecture Thesaurus (aat). Together these collections comprise 22 million RDF triples.

The second search engine is the search facility of the Rijksmuseum web site[6] (Rijks), a Dutch art museum. The log files cover 5.5 months and consist of 216,217 queries in 45,046 sessions, where sessions were identified using IP addresses and agent fields. As linked-data, we used WordNet, Cornetto, the Dutch version of the Getty thesauri, and also various Dutch art-specific ontologies that were collected and interlinked in the E-Culture project[7].

For both data sets the support threshold was set at 0.0005 and the confidence threshold at 0.66667.

4.2 Semantic modification patterns

For 55% of the 482,717 query modifications in the News photo data set and 46% of the 49,410 query modifications in the Rijksmuseum data set, the two queries could both be mapped onto concepts in the linked data (see Table 1). In both data sets a relation was found for about half of the modifications for which concepts were found. On average 12.5 (News photo) and 6.2 (Rijksmuseum) relations were found per query pair.

	News photo	Rijksmuseum
no concept found	0.45	0.54
no relation found	0.30	0.23
relation found	0.25	0.23

Table 1: Proportion of query pairs for which a relation could (not) be found in the linked data.

We manually evaluated the concepts that were found for 100 random queries from the News photo data set. For 74% of

[4]http://www2.let.vu.nl/oz/cltl/cornetto/
[5]http://www.getty.edu
[6]http://www.rijksmuseum.nl/
[7]http://e-culture.multimedian.nl/

Table 2: The 10 semantic modification patterns with the highest support in the News photo data set.

	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	0.86	`Q1 −DBpedia:birthplace→ X ←DBpedia:birthplace− Q2`
5.	0.013	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.	0.008	0.96	`Q1 −aat:distinguished_from→ Q2`
10.	0.008	0.96	`Q1 −WordNet:memberMeronymOf→ X ←WordNet:memberMeronymOf− Q2`

Table 3: The 10 semantic modification patterns with the highest support in the Rijksmuseum data set.

	support	confidence	pattern
1.	0.123	0.96	`[ ]`
2.	0.020	0.74	`Q1 −WordNet:hyponymOf→ X ←WordNet:hyponymOf− Q2`
3.	0.008	0.73	`Q1 −Cornetto:domain→ X ←Cornetto:domain− Q2`
4.	0.003	0.92	`Q1 −rdf:type→ X ←rdf:type− Y ←Rijks:material− Z −Rijks:schilder→ Q2`
5.	0.003	0.87	`Q1 −Cornetto:hasHyperonym→ X ←Cornetto:hasHyperonym− Q2`
6.	0.003	1.00	`Q1 ←Cornetto:hasHyperonym− Q2`
7.	0.003	1.00	`Q1 −WordNet:hyponymOf→ Q2`
8.	0.002	1.00	`Q1 −Cornetto:hasHyperonym→ Q2`
9.	0.002	0.72	`Q1 −Getty:nationalityNonPreferred→ X ←Getty:nationalityNonPreferred− Q2`
10.	0.002	0.71	`Q1 −Cornetto:domain→ X ←Cornetto:domain− Y −Cornetto:eqNearSynonym→ Q2`

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 for 100 random query pairs the relations that were found. 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 appear to be more or less random, so that the patterns that occur in high numbers are in majority based on correct paths.

The support and confidence thresholds effectively removed many irrelevant patterns. An example of a discarded pattern in the News photo data is:

```
Q1 −rdf:type→ X ←rdf:type− Y −DBpedia:
birthplace→ Z ←DBpedia:birthplace− Q2
```

This pattern applies to query pairs consisting of two persons `Q1` and `Q2`, so that there exist some person `Y` who is of the same type as `Q1` (e.g., both tennis players) and is born in the same place at `Q2`. This pattern occurred quite frequently between queries from the same session (support 0.0015), but even more frequently between queries from the different sessions (confidence 0.39) and thus was discarded.

The patterns with the highest support in the News photo data are shown in Table 2. The most common pattern was the identity relation ([ ]): two different queries that are mapped to the same concept, usually variant names for the same entity, such as 'Gent' and 'Gand' (the Dutch and French name of a Belgian City). 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 both say that users searched on two entities from the same type, such as tennis players or townships. Pattern 10 shows that people often search for two entities that are part of the same whole. Inspection of the query pairs that follow this pattern shows that these are mainly entities that are both part of the concept `WordNet:royalty` (e.g., queries `princess` and `king`).

Compared to the News photo data, the Rijksmuseum data shows more patterns that involve concepts named by common nouns (e.g., patterns 2, 5, 6, 7 and 8 in Table 3) and less concepts named by proper nouns (e.g., pattern 9 in Table 3). This difference has consequences for the support that should be offered by the search engines. Users of the Rijksmuseum web site would probably be helped best by showing terms that are related to their current query via relations from a thesaurus. The News photo users, on the other hand, would most likely benefit more from showing entities that are related to their current query via domain specific relations.

Another difference between the Rijksmuseum patterns and the News photo patterns, is that in the Rijksmuseum data

the identity relation ([ ]) has a much higher support. This shows that users of this search engine have more trouble formulating their query and thus try more variant names before finding the right name for the entity they search for. The search engine could support this by offering a list of spelling variants whenever the current query does not yield any results.

In both data sets a large proportion of the relations were *sibling relations*: relations of the form `Q1`–`R`→`X`←`R`–`Q2`. Examples include patterns 4, 5, 6, 8 and 10 in Table 2 and patterns 2, 3, 5 and 9 in Table 3. In the News photo data set these patterns made up 22% of the determined relations; in the Rijksmuseum data set 9%. This shows that many people search for two entities with some common property, such as two actors starring in the same movie or two hyponyms of a WordNet concept. This finding is in line with the work of Rieh and Xie [14], who found through manual classification of query modifications that sibling relations (referred to as 'parallel movements' by them) were the most common modification type. However, their analysis did not identify what kind of siblings were used (actors, or soccer players, or hyponyms, etc.).

Another frequently occurring relation type (14% News photo, 2% Rijksmuseum) are *direct few-to-few relations*, such as 'spouse-of' and 'has-capital'. Here few-to-few relations are defined as a relaxed version of one-to-one relations, where 'few' means on average less than 2.

Finally, we observe that most of the relations that are found (74% News photo, 68% Rijksmuseum) consist of more than one link. Apparently, users often search for entities that are related in a complex way.

4.3 Comparison with term-based analysis

For comparison, we also analyze the data sets with a term-based approach (see Section 2). First, the query terms are stemmed. Then, for each stemmed query we determine whether compared to the previous query terms are added (specification), removed (generalization) or replaced (reformulation). In addition, we count how many times stemming made the query the identical to the previous query (remember from Section 3.1 that consecutive queries that were identical before stemming are removed). Query pairs without overlapping terms are classified as 'undetermined'. The frequency of each type of term modification is shown in Table 4 and Figure 3.

In the News photo data reformulations occur most frequently, followed by specifications, generalizations and stem-identical queries. These findings closely match the findings of previous studies [5, 13, 11, 14, 15, 10, 3]. In the Rijksmuseum data we find the same pattern except that in these data there are a relatively large number of specifications. This distribution is in line with the findings in [7], but it is unclear why it differs from the other data sets.

modification type	News photo	Rijksmuseum
specification	0.08	0.13
generalization	0.05	0.07
reformulation	0.11	0.10
stem-identical	0.01	0.03
undetermined	0.75	0.67

Table 4: Frequency of term-based query modification patterns.

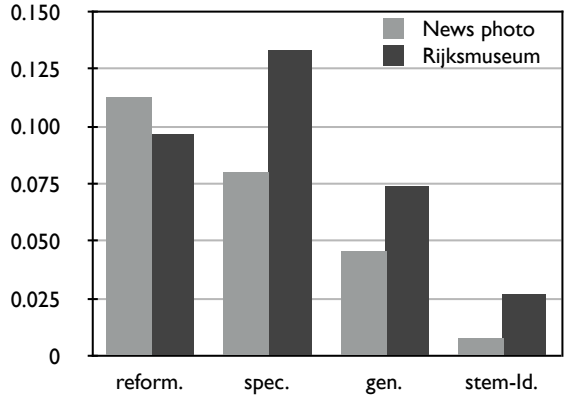

Figure 3: Relative frequency of term-based modification types.

25% of the modifications in the News photo data and 33% of the modifications in the Rijksmuseum data can be assigned to one of the four term-based classes. These percentages are comparable to the percentages of cases that could be assigned to a semantic modification class (see Table 1). However, as shown in Figure 4 the two approaches classify different query pairs: the linked data approach found a relation for only 9% (News photo) and 19% (Rijksmuseum) of the cases that were classified by the term-based approach. Conversely, the term-based approach found a class for only 9% (News photo) and 27% (Rijksmuseum) of the cases for which a semantic relation was found. One reason for this effect is that the term-based approach works well for query pairs consisting of multiple entities, such as `Beatrix`, and `Beatrix holiday`, but cannot handle most pairs consisting of single entities, such as `Beatrix`, and `Willem-Alexander`. The linked-data approach, on the other hand, can handle single entity queries, but not multiple entity queries. This indicates that the two approaches are to a large extent complementary.

Reformulations are related to sibling relations: for example, the modification from the query `elm tree` to the query `oak tree` is both a reformulation and a sibling relation. However, a large part of the siblings cannot be recognized by looking at reformulations: only 6% (News photo) and 2% (Rijksmuseum) of the siblings were recognized as reformulation. Inspection of the results shows that many siblings consist of names of two persons, such as two players in the same national team. The names do not have any terms in common and thus are not classified as reformulations. In other words, the queries were semantically related but not in terms of terms.

There is no corresponding term-based class for direct few-to-few relations. These relations are classified as reformulations, specifications, generalizations and stem-identical, but most often as undetermined.

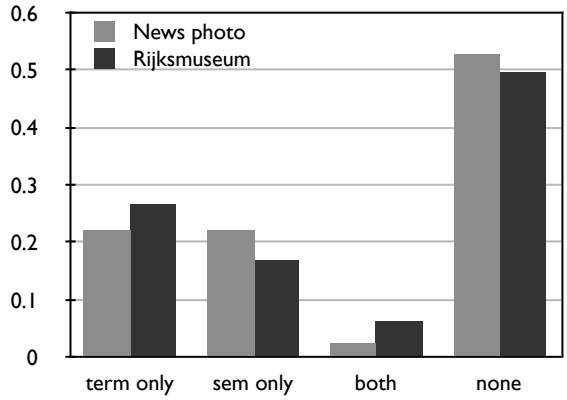

Figure 4: Overlap between the query pairs classified by the semantic and the term-based approach: the proportion of queries for which a relation is found only by the term-based approach, only by the semantic approach, by both approaches, and by none of the approaches

In conclusion, there are types of modifications that appear to be important for users, but that cannot be identified with a term-based analysis of query modifications. Conversely some modifications that can be classified with a term-based approach cannot currently be classified using linked data, most notably modifications involving queries with multiple entities. Thus, the linked data approach does not make a term-based analysis obsolete, but forms a valuable addition to it.

5. CONCLUSIONS

In this paper we showed the potential of combining statistical information gathered from log files with semantic information from linked data. The use of 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, the linked data approach finds relations between queries that do not have any terms in common. This is a large advantage as our analysis indicates that users often search for two entities sharing a common property (e.g., both being tennis players) or two entities with a direct relations (e.g., spouses). These entities are related semantically, but do not have common terms. Moreover, with the linked data approach we can find more detailed modification patterns than with term-based approaches.

Insights gained from semantic query modification analysis can be used directly for improvement of search engines. We found that users often try various names to find the same entity indicating that users are often unsure about which names are used in the data set. Search engines can support this by showing a list of variants of the current query that occur in the data set. Furthermore, we found that users of one search engine mainly modified their queries according to domain-specific relations, such as partner-of, while users of another search engine tended to use thesaurus relations such as hyponym-of. Knowing which types of relations are important for the users of a search engine, enables us to offer search support that is tailored to the user population of that search engine.

One form of search support are suggestions for follow-up queries. In the next step of our research we will apply semantic query modification patterns to query suggestion. If we know that users who search on the name of a soccer player often also want information about other players from the same team, we can suggest the names of other players to a user who searches on the name of one player. In contrast to purely statistical query suggestion methods, these types of patterns can also be applied to queries that are entered for the first-time.

Another direction of further research will be the extension of our analysis from query pairs to query sequences. By examining search sessions as a whole, we can reveal session-wide search patterns that extend beyond individual query pairs.

6. REFERENCES

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

[2] 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.

[3] 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 and Intelligent Agent Technology, Milan, Italy*, pages 183–190, 2009.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] B. J. Jansen. Search log analysis: What it is, what's been done, how to do it. *Library and Information Science Research*, 28(3):407–432, 2006.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.