Deriving Query Suggestions for Site Search*

Udo Kruschwitz, Deirdre Lungley, M-Dyaa Albakour and Dawei Song

November 27, 2012

Abstract

Modern search engines have been moving away from very simplistic interfaces that aimed at satisfying a user's need with a single-shot query. Interactive features such as query suggestions and faceted search are now integral parts of Web search engines. Generating good query modification suggestions or alternative queries to assist a searcher remains however a challenging issue. Query log analysis is one of the major strands of work in this direction. While much research has been performed on query logs collected on the Web as a whole, query log analysis to enhance search on smaller and more focused collections has attracted less attention, despite its increasing practical importance. In this paper, we report on a systematic study on different query modification methods applied to a substantial query log collected on a local Web site that already employs an interactive search engine. The purpose of the analysis is to explore different methods for exploiting the query logs to derive new query modification suggestions. We conducted experiments in which we asked users to assess the relevance of potential query modification suggestions that have been constructed using a range of log analysis methods as well as different baseline approaches. The experimental results demonstrate the usefulness of log analysis to extract query modification suggestions. Furthermore, our experiments demonstrate that a more fine-grained approach than grouping search requests into sessions allows to extract better refinement terms from query log files. Finally, locally collected log files are shown to be potentially useful for extracting term relations that are relevant beyond the domain on which they were collected.

1 Overview

Interactive information retrieval has received much attention in recent years, e.g. (Ruthven, 2008; Marchionini, 2008; Tunkelang, 2009). Evidence for the usefulness of interactive search systems is the fact that faceted search has become popular over recent years, e.g. (Ben-Yitzhak et al., 2008), and prominent Web search engines have added more and more interactive features, including both suggestions as the user types as well as related searches and modification terms following the query submission, e.g., Google, Yahoo! and Bing.

The problem is that traditional Web search is fundamentally different to searches where users are not just interested in getting *some* matching documents but where they are looking for specific documents, memos, spreadsheets, books, etc. Such information requests are not necessarily best served by a single-shot unstructured query. This type of search is very common

*This is a preprint of an article accepted for publication in *Journal of the American Society for Information Science and Technology*

in enterprise search which runs on smaller, often more structured collections (Hawking, 2010). Local Web sites are similar in that they contain much less redundancy than found on the Web and the users' information needs might only be met by a single document in the collection. An example drawn from our sample domain is a user searching for the exam timetable, there is in fact only a single file, an Excel spreadsheet, that contains the official exam timetable.

How can a user be guided in the search process? Library classification schemes like the *Universal Decimal Classification*[1] (UDC) have been used for decades and have been demonstrated to be useful when classifying books. The drawback that these manually encoded classification schemes have is that they lack flexibility. Furthermore, they represent a structured view of the world but that view may not be the view that an *online* searcher has. One possible step towards more interactivity is to improve query modification suggestions proposed by the search engine. It is recognized that there is great potential in mining information from query log files in order to improve a search engine (Jansen, Spink, and Taksa, 2008; Silvestri, 2010). Given the reluctance of users to provide explicit feedback on the usefulness of results returned for a search query, the automatic extraction of implicit feedback has become the centre of attention of much research (Clark et al., 2012). Queries and clicks are interpreted as "soft relevance judgments" (Craswell and Szummer, 2007) to find out what the user's actual intention is and what the user is really interested in. This knowledge is then used to improve the search engine by either deriving query modification suggestions (the focus of our research) or by using such knowledge to modify the ranking of results to make the search system adaptive to the user or the entire user population, e.g. (Joachims and Radlinski, 2007). A simple demonstration of this feedback cycle already being employed is a query submitted to Google from two different computers on the *same* desk. The results returned by the search engine are in many cases different and are likely to reflect the user's past search behaviour (on each of the computers) (Zamir et al., 2005).

Much of the research into search technologies concentrates on internet (Web) search, but less has been reported on enterprise search, domain-specific search or search over individual Web sites which is all different from general Web search (Hawking, 2010; White, 2007; Sherman, 2008). We focus on search over individual Web sites. We use the log files to identify pairs of related queries, which are then employed as query modification suggestions in an interactive search environment. Our focus is not on finding out what specific type of relation holds between the two queries but whether the user perceives the relation as useful in the search context. The motivation for this lies in the observation that users of interactive search engines use suggested terms to give them ideas for what terms are related (in some way) to the original query. The semantic relationships of such feedback terms does not have to be immediately apparent, for example, a Web search query *"worldwide petrol prices"* could trigger a variety of related terms such as *pipe, iraq* and *dollar*, all of which could be good feedback terms, no matter what type of relation they represent (White and Ruthven, 2006). Another motivation for us is the intention to ultimately use the extracted relations to allow the users of a Web site to explore (rather than search) the collection.

In this paper we report on a systematic study on different query modification methods applied to a substantial query log collected on a local Web site. We explore methods that exploit log files as well as a number of methods that extract suggestions from the actual documents. The caveat of such a study is that it is limited to a single Web site and the findings may or may not be transferable to other document collections. Despite this limitation, we see two contributions that our study makes. First of all, it demonstrates how to systematically evaluate query suggestions (prior to assessing them in a live system). Secondly, the systematic evaluation of a comprehensive list of log-based query suggestion methods over a local website is of significance. Our

[1] `http://www.udcc.org/`

results could provide insights and serve as a baseline for future studies on different Web sites. The major bottleneck in conducting research into query logs is the difficulty in getting hold of *realistic* and *large-scale* log data.

The paper is structured as follows. We will start with a brief discussion of related work (Section 2). We present our research questions in Section 3. Section 4 will describe the interactive search engine for which we have collected a query corpus (described in Section 5) that is used in our experiments. In Section 6 we will present the different methods for deriving query suggestions employed in this research. We will then describe the experimental setup (Section 7). Section 8 presents the results followed by a detailed discussion in Section 9. In the conclusions (Sections 10) we will relate the findings to our research questions. We will finish with an outlook on future work in Section 11.

2 Related Work

The idea of supporting a user in the search process by interactive query modifications has been discussed extensively (Efthimiadis, 1996). There is also evidence that integrated interactive IR systems can offer significant advantages over baseline systems (Yuan and Belkin, 2007). Ruthven states however that searchers can have difficulties in identifying useful terms for effective query expansion (Ruthven, 2003). Nevertheless, query suggestions can be useful, and they can help in the search process even if they are not clicked on (Kelly, Gyllstrom, and Bailey, 2009).

Many ideas have been proposed to address the problem of information overload when searching or exploring a document collection. One very promising route is *Social Search* which combines ideas from personalization and social networking so that a searcher can benefit from past users' search experiences (Smyth et al., 2009). The question is what search trails and information should be exploited in this process. Utilizing explicit user judgements about items or search terms seems to be most useful. The problem is however that users are reluctant to leave any explicit feedback when they search a document collection (Dumais et al., 2003; Jansen, Spink, and Saracevic, 2000; Markey, 2007). Instead, implicit feedback, e.g., the analysis of log records, has been shown to be good at approximating explicit feedback. There is a wealth of related work in log analysis, interactive search and other areas (Jansen, Spink, and Taksa, 2008; Silvestri, 2010). For example, users often reformulate their query and such patterns can help in learning an improved ranking function (Joachims and Radlinski, 2007). The same methods have shown to improve query suggestions encoded in an adaptive domain model on a local Web site (Lungley and Kruschwitz, 2009). Log analysis has in fact developed into an entire research strand and it has been widely recognised that query log files represent a good source for capturing implicit user feedback (Jansen, Spink, and Taksa, 2008; Silvestri, 2010) or evaluating the retrieval function (Radlinski, Kurup, and Joachims, 2008).

The next question is how such feedback should be applied to improve the search process. One possibility is to exploit it in order to build knowledge structures that can assist in interactive search. But do users want assisted search? There is evidence that users want support in proposing keywords but they ultimately want to stay in control about what is being submitted as a query (White and Ruthven, 2006). Furthermore, despite the risk of offering irrelevant suggestions in system-guided interactive search, users might prefer having them rather than not (White, Bilenko, and Cucerzan, 2007). Within the context of local Web site search it was also found that users want such support (Kruschwitz and Al-Bakour, 2005).

Belkin calls the move beyond the limited, inherently non-interactive models of IR to truly interactive systems the *challenge of all challenges* in IR at the moment (Belkin, 2008). Our

primary interest is in one specific aspect of this challenge, namely how to best model query modification suggestions.

There are many different ways of structuring such models. Models can be built by extracting term relations from documents, e.g. (Sanderson and Croft, 1999; Kruschwitz, 2005; Widdows and Dorow, 2002), anchor text pointing to documents (Kraft and Zien, 2004) or from the actual queries that users submit to search the collection by building query flow graphs, e.g. (Boldi et al., 2008; Boldi et al., 2009; Deng, King, and Lyu, 2009), or mining term association rules (Fonseca et al., 2004). Log files are a promising source for this purpose since past user queries appear to be preferred by users when compared to terms extracted from documents (Kelly, Gyllstrom, and Bailey, 2009) and more recently it has been shown that combining them both will further improve the retrieval effectiveness (Adeyanju et al., 2012). Various Web log studies have been conducted in recent years to investigate the users' search behaviour, e.g. (Anick, 2003; Wang, Berry, and Yang, 2003; Chau, Fang, and Sheng, 2005; Jansen, Spink, and Koshman, 2007), and log files have widely been used to extract meaningful knowledge, e.g. relations between queries (Baeza-Yates and Tiberi, 2007), or to derive query substitutions (Jones et al., 2006). Also there have been various attempts to understand query reformulation strategies in Web search and evaluate their effectiveness (Huang and Efthimiadis, 2009; Liu et al., 2010). Much of this work however is based on queries submitted on the *Web* and thus presents a very broad view of the world. Our work is different in that we start with a specific document collection for which suitable knowledge structures are typically not readily available (such as a local Web site), and generate query suggestions extracted from queries previously submitted within this collection using a range of methods for (a) linking related queries and (b) turning the queries into query suggestions.

Looking at query suggestion methods that work effectively outside the general Web domain is now becoming more popular, e.g. (Al Hasan et al., 2011). Our earlier studies with individual methods for extracting query suggestions have shown promising results, e.g. (Kruschwitz et al., 2011; Dignum et al., 2010; Albakour et al., 2011a). Here we investigate a more comprehensive list of different methods in a more systematic way.

3 Research Questions

Given the popularity of interactive search suggestions we try to explore how to best exploit the interaction patterns collected on search engines of local Web sites to offer query suggestions. The idea is to maximise the usefulness of suggestions proposed by the system. Following on from the discussion of related work and the issues arising from it, the research questions we aim to answer with our experiments are as follows:

1. *Are log files of an interactive local search system an appropriate resource to derive query modification suggestions for site-specific search?* We will address this question through a systematic study on different query modification methods applied to a substantial query log, and compare those with several baseline methods that do not use log data. As part of this question we also investigate how different approaches of segmenting search sessions might affect the relevance of the derived query suggestions.

2. *Is the relevance of such feedback terms perceived differently by searchers on the local Web site compared to the general Web population?* This question tries to find out whether the extracted suggestions are primarily site-specific or more generic than that.

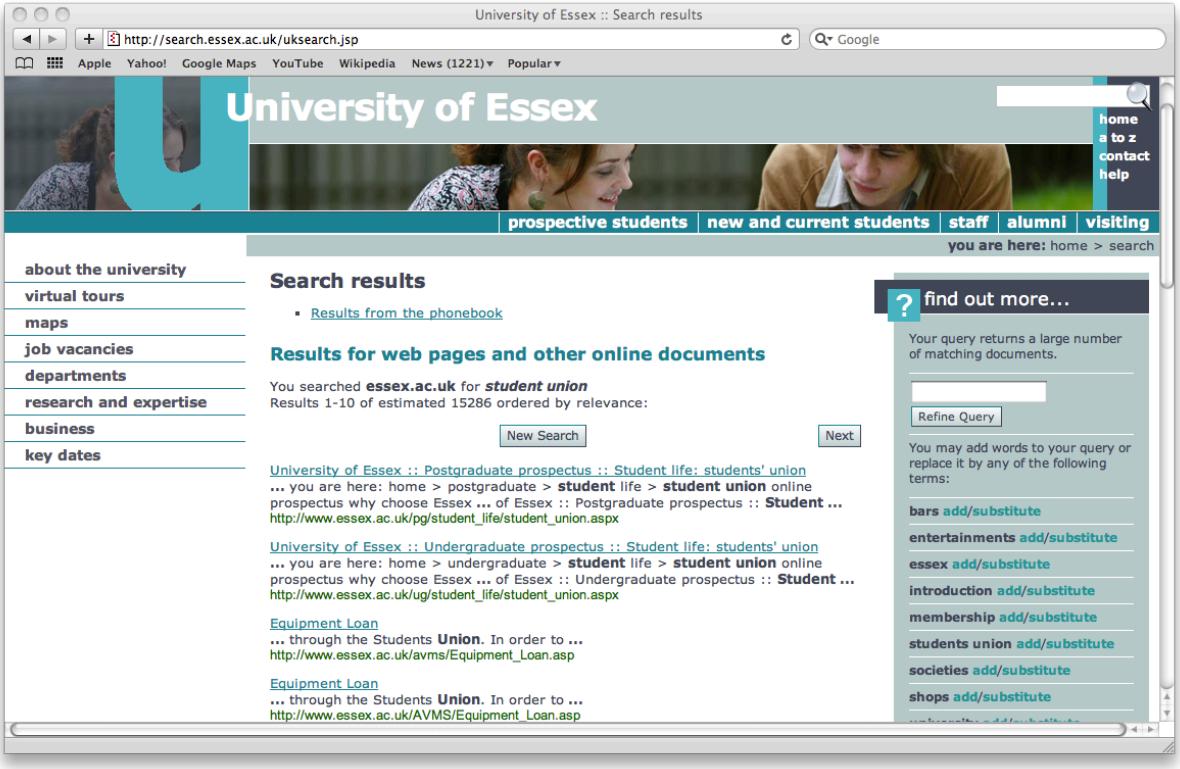

Figure 1: System response to user query *"student union"*

4 Search System Setup

For this study we focus on the search engine that has been running on the Web site of the University of Essex[2] for more than three years (Kruschwitz, Webb, and Sutcliffe, 2008). The search engine already employs an element of interactivity similar to query suggestions proposed by state-of-the-art Web search engines.

We use *Nutch*[3] as the backend search engine. The crawler works like that of a standard Web search engine. It starts with a small number of seed pages and explores the link structure within the domain accordingly. An interactive system is built upon the backend search engine to respond to a user query by presenting the top matching documents along with some suggestions for query modification. The output looks like that shown in Figure 1 (a sample screenshot of the system following a frequent user query *"student union"*).

We know that queries submitted to search engines are typically short, normally between two and three words long, e.g. (Silverstein, Henzinger, and Marais, 1998; Beitzel et al., 2007), and the majority of queries result in a large set of matching documents even in small collections, e.g. (Kruschwitz, 2003). We assume that a large proportion of user queries can be answered by inspecting the documents returned by a state-of-the-art search engine. However, we also observe that there is a percentage of queries that cannot be answered with a one-shot query be it that they are ambiguous or very generic. Query modification suggestions derived in the context of this work are primarily aimed at assisting searchers who submit these queries.

For each search request, the system derives query modification suggestions from both an

[2] `http://search.essex.ac.uk`
[3] `http://lucene.apache.org/nutch/`

automatically acquired domain model and the best matching documents. Specifically, our domain model is a term association graph. It is constructed in an automated offline process and exploits the markup structure found in the documents of the entire collection and is explained in more detail in (Kruschwitz, 2005).

For the extraction of terms from matching documents, we use the titles and snippets returned by the search engine. We assign parts of speech and select nouns and certain noun phrases (the idea is to use patterns that can identify collocations in documents). We consider nouns and noun phrases to be the most useful phrases for retrieval tasks. For the detection of noun phrases we look for particular patterns, i.e. sequences of part-of-speech tags based on the algorithm for the detection of terminological terms described in (Justeson and Katz, 1995). Patterns of length two and three form the vast majority of terminological terms according to Justeson and Katz. There are two admissible patterns of length two and five of length three as can be seen in Table 1 (where A is an adjective, P is a preposition and N is a noun). The examples are drawn from the *University of Essex* sample domain and converted into a normalised form.

Pattern	**Examples**
A N	*disciplinary committee*
N N	*student support*
A A N	*third international conference*
A N N	*undergraduate degree schemes*
N A N	*research active staff*
N N N	*albert sloman library*
N P N	*department of economics*

Table 1: Part-of-speech patterns for interesting noun phrases

Finally we select the most frequent nouns and noun phrases we identified and add them to the refinement terms suggested by the domain model. We display up to 10 terms derived from the domain model followed by the (up to 20) most frequent ones calculated on the fly. Suggestions are derived from the domain model by identifying a term association hierarchy in the model whose root node contains the query, outgoing links (to related terms) are rank-ordered and selected as suggestions as explained in more detail elsewhere (Kruschwitz, 2005).

In the current system no learning mechanism is involved when deriving query suggestions for a submitted query.

5 User-System Interactions

In this work we primarily explore query logs to generate query suggestions. We assume a fairly generic structure of the log files. No clickthrough information is exploited. However, apart from simply recording user queries with an associated session identifier we also assume that some basic features present in an interactive search engine are recorded as part of the query logs. Figure 1 illustrates that an initial query is not simply answered by a set of matching documents, but the searcher is offered suggestions for query modification that can include query terms to be added to the current query or to substitute the query. There is also a text box that allows the user to type in some new search terms (to either replace the previous query or modify it depending on the given context).

The Essex University Web site we investigated currently consists of about 70,000 indexed pages. We have collected query logs comprising more than 2,000,000 queries, a substantial

proportion of the interactions with the system following on from an initial query. The log files are an extremely valuable resource because they are a reflection of real user interests when searching the university Web site.

5.1 Query Logs

Here is an extract from the actual log files to illustrate the structure ('`xxx`' is a field separator):

```
...
1715390 xxx 890C463BD77BF9A3F1BBCE8F2C38A8B8 xxx Sun Dec 12 13:23:03 GMT 2010 xxx 0 xxx 0 xxx 0\
            student union xxx student union xxx Student Union
1715391 xxx 890C463BD77BF9A3F1BBCE8F2C38A8B8 xxx Sun Dec 12 13:23:20 GMT 2010 xxx 0 xxx 0 xxx 0\
            students union xxx students union xxx Students union
...
```

The logs record a query identifier, a session identifier, the submission time, various forms of the submitted query as well as additional information (explained below). Displayed are two interactions submitted within the same session. The query *"Student union"* is followed by a new query: *"Students union"*.

The sample log entries also demonstrate that we do not identify individual users and we do not associate IP addresses with sessions. The underlying reason for that is to comply with data protection issues and to avoid any potential privacy problems. It also fits in with our overall aim of generating suggestions for an entire community of users which is different from personalised suggestions.

5.2 Sessions

The second field in the log record is the automatically generated session identifier. Automatically identifying the boundaries of sessions is a difficult task (Göker and He, 2000; Jansen et al., 2007). One of the reasons is that a session can easily consist of a number of *search goals* and *search missions* (Jones and Klinkner, 2008). Identifying topically related chains in user query sessions has been studied extensively (Gayo-Avello, 2009). We use the default server timeout, i.e. a session expires after 30 minutes of inactivity, a method that has been shown to give highly accurate session boundaries (Jansen et al., 2007).

5.3 Dialogues

Standard query log analysis breaks user interactions into sessions based on a session identifier. However, since we are exploring the log files of an interactive search engine we can decompose individual sessions into more fine-grained interactions as indicated earlier. Here is another short extract from the log files we use:

```
...
1648403 xxx 942D437CE08F5016514DD550FF188DC8 xxx Sat Nov 13 18:34:32 GMT 2010 xxx 0 xxx 0 xxx 0\
            student union xxx student union xxx student union
...
1648405 xxx 942D437CE08F5016514DD550FF188DC8 xxx Sat Nov 13 18:34:51 GMT 2010 xxx 1 xxx 1 xxx 0\
            bars xxx student union  bars xxx NID3555bars<a>
...
```

In this case the initial user query is *"student union"*. The user then selects a suggestion made by the system. The user adds the term *"bars"* (cf. Figure 1) to the inital query in a query refinement step. The query logs record that the new query is part of an existing interaction

and we define any such interactions as a *dialogue*. More precisely, if a user either selects a suggested term (for refinement or replacement) or if a user submits a follow-up query as part of the interaction (e.g. see the "Refine Query" button in Figure 1), then we treat such interactions as part of a *dialogue*. The dialogue continues as long as the user keeps selecting terms proposed by the system or modifying the query via the appropriate text box until the user starts a new search in the initial search box or the session expires.

In summary, the log files of an interactive search engine can be used to not just group queries that were submitted within the same session, but also to define a more fine-grained grouping that we call a dialogue from now on. We shall now discuss the different techniques for extracting search suggestions investigated in this study.

6 Deriving Query Modification Suggestions

To acquire query modification suggestions for a given user query we explore a variety of methods. Figure 2 illustrates that we distinguish methods that either derive query suggestions from previously submitted queries (using logs) or from the actual documents (our baseline approaches). Log-based methods are further divided into adaptive and non-adaptive methods. Adaptive in this context means that the log data is processed incrementally (according to the method at hand), and these methods can be seen as building a continuously updated "domain model" representing queries and query suggestions. The underlying idea is that adaptive methods will be able to learn over time in a continuous learning cycle. Changing the frequency of the update is then likely to result in different models. On the other hand, what we define as non-adaptive methods are methods which simply take the entire log and derive suggestions from the aggregated data, and building these models incrementally would not resut in a different model.

We employ three non-adaptive approaches to acquire query suggestions: Maximum Likelihood Estimation (MLE), Association Rules (AR) and Query Flow Graphs (QFG); non-adaptive, because we derive maximum likelihood estimates, association rules as well as query flow graphs from the entire log at once, and even an incremental construction of these models will not change the final result. We employ a model building approach following the Ant Colony Optimisation (ACO) paradigm as an adaptive approach in our definition. The non-log-based methods extract query suggestions using snippet text (SNIPPET) or word n-grams (NGRAMS). In other words, the baseline approaches both exploit a document collection rather than log data, but the SNIPPET approach selects suggestions from best-matching documents whereas NGRAMS utilise n-grams derived from an entire document collection.

For the log-based methods we will furthermore vary the data preparation process in that we feed the models with either session-based or more fine-grained, dialogue-based data derived from the query logs.

All methods produce a ranked list of suggestions for a given query so that a cut-off point can be defined depending on how many suggestions should be selected. Note that this list may well be empty depending on the actual query at hand. In our experiments we will use a cut-off point of three. In the discussion below where we introduce each method in detail we describe how to select the highest ranked suggestion for each particular method.

By considering the different variations of each technique we end up with a total of 13 different methods to explore query suggestion modifications.[4]

[4]Initially, we also explored an additional baseline, namely suggestions provided by Google (using "Related Searches" under "Show options") when submitting q to Google and specifying `site:essex.ac.uk` in the query. However, it turns out that this would be an unfair comparison as the suggestions are not tailored to the site

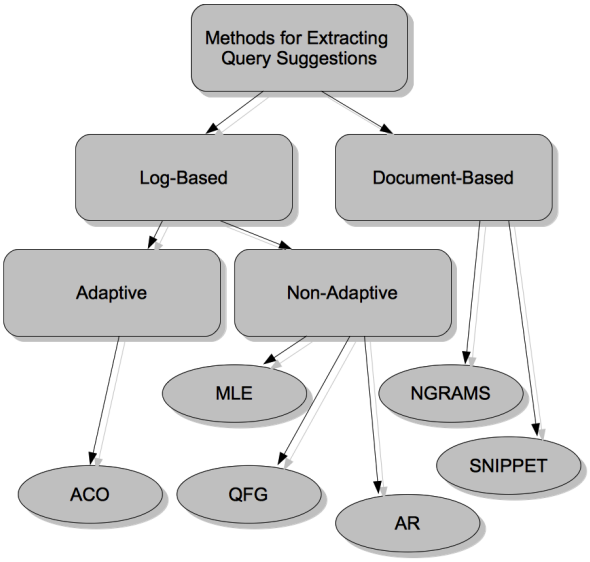

Figure 2: Methods for extracting query suggestions

In order to normalise the query corpus and snippets we performed case folding, i.e. all capital letters were transformed into small letters. We also replaced all punctuation marks such as periods, commas and hyphens by white space.

We will now look into each method in detail.

6.1 Maximum Likelihood Estimation (MLE)

MLE is a commonly used baseline approach in statistical natural language processing (Manning and Schütze, 1999). For each query q, we apply MLE to pairs of queries (extracted from the query modification sequences in the log file), i.e. we calculate $max\ P(q_{n+1}|q_n)$ over all follow-up query pairs $\langle q_n, q_{n+1}\rangle$ observed in the logs. Formally, we select a query q' as suggestion for query q_n such that:

$$P_{MLE}(q'|q_n) = max\ P(q_{n+1}|q_n) \tag{1}$$

$$P(q_{n+1}|q_n) = \frac{freq(\langle q_n, q_{n+1}\rangle)}{freq(q_n)} \tag{2}$$

We distinguish four different ways of defining a valid query modification sequence. The session-based approach considers every pair of queries within a chronologically ordered session as a valid query modification sequence. A more fine-grained approach is to restrict this to

specified in the query.

queries only submitted within the same dialogue (see Section 5). Finally, we investigate the MLE method where the valid query modification pairs provided as input are all those steps within the same dialogue where a user either *added* to or *replaced* the current query. This results in the following four methods to derive query suggestions using MLE:

- **MLE-Session**: the query q' that is most likely to follow q_n within a session.
- **MLE-Dialogue**: the query q' that is most likely to follow q_n within a dialogue.
- **MLE-Dialogue-Add**: the query q' that is most likely to be used as a *query refinement*, i.e. added to q_n, within a dialogue.
- **MLE-Dialogue-Replace**: the query q' that is most likely to be used as a *query replacement*, i.e. substituting q_n, within a dialogue.

Any query pair in the ranked list generated by MLE that occurs only once in the log file will be discarded.

6.2 Association Rules (AR)

Association rules have their roots in the database community and have also been used to derive query suggestions. We implemented Fonseca *et al.*'s approach because it has been shown to work well on Web log data (Fonseca et al., 2003; Fonseca et al., 2004). The general intuition is that if distinct queries co-occur in many sessions, then this may be an indication that these queries are related. Their approach maps the problem of finding association rules in customer transactions to the problem of finding related queries in a Web search engine. In this context, an association rule is an expression $A \rightarrow B$, where A, B are sets of transactional items. The support of the rule is given as $\sigma(A \cup B)$, and the confidence as $\frac{\sigma(A \cup B)}{\sigma(A)}$, in other words the conditional probability that a transaction contains B given that it contains A. Following this, transactional items can be considered as queries A, B and the support $\sigma(A \cup B)$ is the number of sessions that contain both queries. The confidence value of the association rules can then be used to rank suggestions for a given query.

In line with (Fonseca et al., 2004) we use a minimum support of 3, i.e., distinct queries that co-occur less than 3 times are deemed as being unrelated. We employ the same heuristics used by Fonseca *et al.* to delete less useful suggestions, namely:

1. Suggestions that are plural forms of the original query, e.g., the suggestion "timetables" for the query "timetable".
2. Suggestions that are substrings of the original query, e.g., the suggestion "timetable" for the query "exam timetable".

In line with the other log-based methods we distinguish two variants to derive a query suggestion for a query *q*:

- **AR-Session**: the highest-ranked query related to *q* extracted from the log files using Fonseca's association rule mining method applied to sessions.
- **AR-Dialogue**: the highest-ranked query related to *q* extracted from the log files using Fonseca's association rule mining method applied to dialogues.

6.3 Query Flow Graphs (QFG)

We use query flow graphs as introduced by Boldi *et al.* and applied to query recommendations (Boldi et al., 2008). The query flow graph G_{qf} is a directed graph $G_{qf} = (V, E, w)$ where:

- V is a set of nodes containing all the distinct queries submitted to the search engine and two special nodes s and t representing a *start state* and a *terminate state*;
- $E \subseteq V \times V$ is the set of directed edges;
- $w : E \rightarrow (0..1]$ is a weighting function that assigns to every pair of queries $(q, q') \in E$ a weight $w(q, q')$.

The graph can be built from the search logs by creating an edge between two queries q, q' if there is one session in the logs in which q and q' are consecutive.

The weighting function of the edges w depends on the application. Boldi *et al.* developed a machine learning model that assigns to each edge on the graph a probability that the queries on both ends of the edge are part of the same chain. The chain is defined as a topically coherent sequence of queries of one user. This probability is then used to eliminate less probable edges by specifying some threshold. For the remaining edges the weight $w(q, q')$ is calculated as:

$$w(q, q') = \frac{freq(q, q')}{\Sigma_{r \in R_q} freq(q, r)} \tag{3}$$

Where:

- $freq(q, q')$ is the number of the times the query q is followed by the query q';
- R_q is the set of identified reformulations of query q in the logs.

Note that the weights are normalised so that the total weight of the outgoing edges of any node is equal to 1.

The query flow graph can be used for query recommendation by ranking all the nodes in the graph according to some measure which indicates how reachable they are from the given node (query). Boldi *et al.* proposed to use graph random walks for this purpose and reported the most promising results by using a measure which combines relative random walk scores and absolute scores. This measure is

$$\overline{s}_q(q') = \frac{s_q(q')}{\sqrt{r(q')}} \tag{4}$$

where:

- $s_q(q')$ is the random walk score relative to q i.e. the one computed with a preference vector for query q (the random walker starts at the node q);
- $r(q')$ is the absolute random walk score of q' i.e. the one computed with a uniform preference vector (equal probabilities of starting the random walk at any node).

In our experiments, we adopt this measure for query recommendation and use the random walk parameters reported by Boldi *et al.* Two variants are used in this study:

- **QFG-Session**: In this case, we build the edges on the graph by considering only subsequent queries in the same session.
- **QFG-Dialogue**: In this case, we build the edges on the graph by considering only subsequent queries in the same dialogue.

6.4 Ant Colony Optimisation (ACO)

Ant Colony Optimisation has been studied extensively as a form of swarm intelligence technique to solve problems in several domains such as scheduling (Socha, Sampels, and Manfrin, 2003), classification (Martens et al., 2007) and routing problems in telecommunication (Di Caro and Dorigo, 1998). Here, the ACO analogy is used to first populate and then adapt a directional graph similar to Query Flow Graphs. In this analogy the edges in the graph are weighted with the pheromone levels that the ants, in this case users, leave when they traverse the graph.

The user traverses a portion of the graph by using query refinements (analogous to the ant's journey), the weights of the edges on this route are reinforced (increasing the level of pheromone). Over time all weights (pheromone levels) are reduced by introducing some evaporation factor to reflect unpopularity of the edge if it has not been used by ants. In other words, we reduce the weight of non-traversed edges over time, to penalise incorrect or less relevant query modifications. In addition we expect outdated terms to be effectively removed from the model, i.e., the refinement weight will become so low that the term will never be recommended to the user.

Let us assume that we update the pheromone levels on a daily basis. For the edge $q_i \rightarrow q_j$ the pheromone level w_{ij} is updated using Equation 5.

$$w_{ij} = N * ((1 - \rho)w_{ij} + \Delta w_{ij}) \tag{5}$$

where:

- N is a normalisation factor, as all pheromone trail levels are normalised to sum to 1;
- ρ is an evaporation co-efficient factor;
- Δw_{ij} the amount of pheromone deposited at the end of the day for the edge $q_i \rightarrow q_j$. The amount of pheromone deposited should correspond to ant moves on the graph. In our case, this can be the frequency of query refinements corresponding to the edge. Also the cost of ant moves can be taken into account when calculating the amount of pheromone deposited. Generally it can be calculated using Equation 6 (Dorigo, Birattari, and Stutzle, 2006).

$$\Delta w_{ij} = \Sigma_k Q/C_k; \text{ For all ant moves on edge } q_i \rightarrow q_j \tag{6}$$

  where:

  - Q is a constant,
  - C_k is the cost of ant k journey when using the edge $q_i \rightarrow q_j$.

In previous work (Albakour et al., 2011a; Albakour, 2012), we experimented with a number of evaporation factors for Equation 5 and pheromone deposition calculation schemes in Equation 6. Based on the finding of those studies, we used the best-performing combination of the parameters in which the evaporation factor ρ = 0.1 and only immediate refinements in the sessions are considered to update the pheromone level on the graph. In this work, the constant Q in Equation 6 is chosen to be the average weight of all edges in the graph in the previous day, and the cost is considered to be the distance between the queries in the session (for immediate refinements $C = 1$).

After updating the graph from the logs, the query recommendation is done by simply listing the connected nodes to the query in question and ranking them by the pheromone levels. The two version of the ACO approach used in this study are:

- **ACO-Session**: In this case, we build the edges on the graph by considering only subsequent queries in the same session.
- **ACO-Dialogue**: In this case, we build the edges on the graph by considering only subsequent queries in the same dialogue.

6.5 Result Snippets (SNIPPET)

Snippets have been shown to be a good source for query expansion terms, e.g. (Yin, Shokouhi, and Craswell, 2009). This is why we include suggestions derived from result snippets as a baseline method. This method selects the most frequent term extracted from the top ten snippets returned for *q*, exploiting part-of-speech patterns using the same snippet processing method as outlined in Section 4 applied to the results returned by the interactive search engine. In order to make this baseline not artificially weak, we treat a number of terms as domain-specific stopwords which will not be selected, namely *university of essex*, *university* and *essex*. Furthermore, we choose the longest compound if several terms are equally frequent (e.g., for the query *"library"*, we choose *"albert sloman library"* as a suggestion since it is as frequent as *"albert"*, *"sloman library"* and *"sloman"*).

For all terms extracted from the top snippets returned by query q (resulting in a set of terms T_q) extract t as query suggestion such that:

$$P_{SNIPPET}(t|q) = max\ \frac{freq(t)}{\Sigma_{t \in T_q} freq(t)} \tag{7}$$

Our baseline methods are not affected by the session segmentation (as used for the log-based methods). Therefore, we employ a single variant of this method only.

6.6 Word n-Grams (NGRAMS)

Another non-log-based baseline to extract query suggestions is to apply MLE to the document corpus by selecting the word w most likely to be following a given query in the document collection (ignoring stop words), so that for queries of length one we are essentially considering bigram patterns. Hence, we select w as query suggestion for q_n such that:

$$P_{NGRAMS}(w|q_n) = max\ P(w_{n+1}|q_n) \tag{8}$$

$$P(w_{n+1}|q_n) = \frac{freq(\langle q_n, w_{n+1} \rangle)}{freq(q_n)} \tag{9}$$

We distinguish two document collections to derive n-gram suggestions:

- **NGRAMS-Site**: most frequent n-gram using the collection indexed by the Essex search engine.
- **NGRAMS-MSN**: most frequent n-gram using the Microsoft Web n-gram corpus introduced in (Wang et al., 2010). We used the n-grams generated from the body content of Web pages.

7 Experimental Setup

To assess the quality of query modification suggestions we adopted an evaluation strategy proposed in the literature where users had to judge the quality of a term in the context of a given query (Sanderson and Croft, 1999). In line with previous experiments (Albakour et al., 2011b; Dignum et al., 2010) we selected for each query and each of the 13 methods the three best (highest weighted) suggested query modifications. A motivating factor to do so is the observation that users are much more likely to click on the top results of a ranked list (of search results) than to select something further down (Joachims et al., 2005) and it seems reasonable to assume that such a preference is valid for query modification suggestions as well, so we are interested in the top three suggestions only.

An online form was prepared, and subjects were asked to judge whether they considered a particular query suggestion relevant or not for a given query. It could be argued that *related* might be a more appropriate term than *relevant*. Sanderson and Croft deliberately avoided the term *related* as they felt that "judging the relatedness of terms was not possible unless one examined the document texts" (Sanderson and Croft, 1999). They decided to ask whether the relations are *interesting*. We opted for the term *relevant* as we look at it from the perspective of a user searching a particular Web site (rather than the Web in general) and it is in line with the experimental setup of our previous studies, e.g. (Kruschwitz, 2003).

To address the second research question we sampled two sets of subjects. We recruited local users of the search engine and we also used a crowdsourcing platform to obtain results from a sample of the general Web population.

7.1 Sampling Queries

In similar work sampling queries for assessment was done by eliminating very frequent or very infrequent queries (Boldi et al., 2009). However, the assessments were conducted in the context of a Web search engine. In our case we are dealing with a local search engine, frequent queries are important and not necessarily navigational. We also did not exclude infrequent queries to test our method on a wide variety of queries including those in the long tail.

We used a three-year query log file (20 November 2007 till 19 November 2010) of more than 1.6 million queries. We randomly sampled 25 frequent queries as well as 25 queries from the entire query corpus. To select frequent queries we sampled from the *set* of top 50 most frequently submitted queries. When sampling from all queries there were four queries that are also drawn from the frequent set which we ignored so that in total we ended up with 46 unique queries (see Table 2 where the first 25 queries are the ones sampled from the frequent set).

We would like to point out that the sampled queries represent a large proportion of all queries submitted to the search engine. Queries submitted to search engines approximately follow the power-law distribution (Baeza-Yates and Saint-Jean, 2003). Therefore, the relatively small sample of frequent queries represents about 15% of the entire query corpus, i.e. about every seventh query submitted to the search engine over a period of three years.

Table 3 presents the suggestions derived for sample query "accommodation" using all 13 different methods. Table 4 illustrates that in particular the less frequently submitted queries are affected by data sparsity. Query "erol" (presumably a misspelling of "enrol") does result in far fewer suggestions, in fact some methods do not return any. We will discuss this issue in more detail later on.

ID	Query	ID	Query	ID	Query
1	moodle	17	law	33	linda morrison
2	library	18	registry	34	insearch
3	timetable	19	exam timetable	35	printer credits
4	cmr	20	mba	36	towers bedroom
5	enrol	21	email	37	summer storage luggage
6	accommodation	22	study abroad	38	alfonso torrejon
7	ocs	23	nursing	39	registration
8	accomodation	24	health centre	40	linda gossett
9	graduation	25	myessex	41	index
10	psychology	26	frontrunners	42	room map
11	timetables	27	staff research	43	sc111
12	term dates	28	viviene jamescook	44	partner university
13	courses	29	report incident	45	technician
14	student union	30	phd	46	accomdation fees
15	fees	31	foundation degree		
16	sports centre	32	erol		

Table 2: Sampled queries.

Method		Suggestion 1	Suggestion 2	Suggestion 3
Log-Based	Session			
	MLE	accommodation office	fees	accommodation services
	QFG	accommodation information	accommodation services	accommodation office
	AR	accomodation	accomadation	fees
	ACO	accommodation office	fees	accommodation services
	Dialogue			
	MLE	accommodation services	accommodation information	fees
	MLE-ADD	colchester	fees	accommodation information
	MLE-REPLACE	accommodation services	accommodation information	fees
	QFG	accommodation information	accommodation services	private sector
	AR	accomodation	advert	accommodation services
	ACO	accommodation office	accommodation services	fees
Doc.-B.	SNIPPET	directory programme specifications	accommodation administration	computing conferences
	NGRAMS-Site	administration	office	registration
	NGRAMS-MSN	hotels	hotel	search

Table 3: Query suggestions derived for query *"accommodation"*.

7.2 Subjects

Different sets of subjects were recruited as follows:

- **Local Users**: Given the domain-specific context of this research, we first recruited subjects, who are either staff or students at the University of Essex, i.e. typical users of the local search engine, 21 subjects in total.

Method		Suggestion 1	Suggestion 2	Suggestion 3
Log-Based	Session			
	MLE	enrol		
	QFG	enrol	entro	professionalism
	AR	enrol	enroll	postgraduate prospectus
	ACO	enrol	entro	enroll
	Dialogue			
	MLE	enrol		
	MLE-ADD			
	MLE-REPLACE			
	QFG	enrol	course enrolment	module enrolment
	AR			
	ACO			
Doc.-B.	SNIPPET	ibm thinkpad introduction	ibm thinkpad linux	
	NGRAMS-Site	kulahci	alkan	
	NGRAMS-MSN	alkan	shopping	gelenbe

Table 4: Query suggestions derived for query *"erol"*.

- **Web users**: Using *CrowdFlower*[5] to access Amazon's *Mechanical Turk* platform[6], we requested judgments from 20 workers for all the suggestions of each query. It could be argued that this is not a representative sample of Web users, but it has been demonstrated that results aggregated from Mechanical Turk workers approximate expert judgement for a variety of tasks, e.g. (Snow et al., 2008; Callison-Burch, 2009; Albakour, Kruschwitz, and Lucas, 2010).
- **Expert:** In addition, we asked an independent domain expert to conduct the same task. This expert is a member of the technical staff that run the local Web site.

7.3 Task

Experiments with local subjects were conducted in a one-on-one setting. Each subject was asked to fill in an online form as used in (Kruschwitz, 2003). The (written) introduction was the following (with a link to the form that contained the selected pairs of related terms):

> *You are the user of a new search engine which searches the University of Essex intranet. In addition to returning the best matching documents for any given query, this search engine also returns a set of words or phrases (called terms)*
>
> - *that give an indication of what the retrieved documents are about, and*
> - *which can be added to the query in order to refine the search or replace the original query*
>
> *The form below gives a list of term pairs. For each pair, imagine the first term was your original query, and that the second is one of the terms proposed by the search system, which you could use to refine or replace the search. Please judge for each pair whether you think the second term is:*

[5] http://www.crowdflower.com/
[6] http://www.mturk.com/

- *relevant (tick "Yes")*
- *not relevant (tick "No")*

If you do not know, then tick "Don't know".

Here, "relevant" means that you can imagine a situation where the second term is an appropriate refinement or replacement of the query given by the first term.

When considering relevance, remember the particular document collection that is being searched.

Subjects were not told that various different techniques have been used to generate these term pairs. The form contained a list of all *unique* term pairs in random order. It took about one hour to conduct each experiment (including two short breaks). Subjects were paid £10 each.

In the instructions for the Web users we removed the reference to the University of Essex intranet (changing the introductory sentence to "You are the user of a new Web search engine") and deleted the last sentence.

The total number of suggestions generated by the 13 different query recommendation systems was 1,520, out of which 824 were unique suggestions judged by each user. The number of unique suggestions for each query varies between 28 unique suggestions for the query "health centre" and only 3 suggestions for the query *"viviene jamescook"*.

8 Results

First of all we need to decide on an appropriate metric in order to discuss the results. This metric needs to reflect that there is a trade-off between methods that are able to produce a lot of query suggestions (some of which might not be considered relevant) and methods that return very few (but possibly more relevant ones). To obtain a balanced picture we will assess the quality of the suggestions produced by the various recommendation systems using two different metrics. The first metric is *Mean Precision at 3* ($MP@3$). For each query, we look at the judgements provided by a single user and assess the precision of the recommendations provided by the system in question for that query (i.e. the number of suggestions judged as *relevant*). Since we only take up to three recommendations the cutoff point is 3.

$MP@3$ will penalise a method that only proposes a single suggestion for a given query, no matter whether that suggestion is relevant or not. We apply a second metric that we call $MP@Max$ which is the probability that a suggestion is considered relevant by the user. It is the percentage of suggestions judged relevant by the users to the total number of suggestions *produced by the method and assessed* by the users. This metric will therefore *not* penalise a method that makes fewer than three suggestions for a query. For example, assume a method provides a single suggestion for a particular query and this suggestion is judged to be relevant, then this would imply $MP@3 = \frac{1}{3}$ and $MP@Max = 1$. Note that our performance measures do not take into account the rank order of suggestions.

We will first present the results obtained from this study using the metrics introduced above and will then discuss them in more detail in the next section.

Table 5 presents the results for the full query sample. Tables 6 and 7 give the results for the 25 sampled frequent queries and 25 queries sampled from the full log file, respectively. The top-performing systems are highlighted in bold. For local users we also applied pairwise two-tailed t-tests for significance testing (applied to the results obtained by both metrics). Some results (for top-performing systems) are illustrated in the tables.

The overall percentage agreement among the local users stands at 79%. Examples of pairs attracting full positive agreement between the assessors are:

mba ⇒ *masters in business*
email ⇒ *webmail*
psychology ⇒ *psychology bsc*

Two example pairs with indecisive judgements by the users are:

psychology ⇒ *sociology*
law ⇒ *llb*

Method		Local Users		Web Users		Expert	
		$MP@3$	$MP@Max$	$MP@3$	$MP@Max$	$MP@3$	$MP@Max$
Log-Based	Session						
	MLE	0.5072	0.6796	0.5304	0.7064	0.5870	0.7864
	QFG	**0.6225** •	0.6225	**0.7157**	0.7157	0.5870	0.5870
	AR	0.5421	0.5799	0.6021	0.6447	0.5870	0.6279
	ACO	0.5714	0.6309	0.6168	0.6825	0.6014	0.6640
	Dialogue						
	MLE	0.4928	0.7556	0.5051	0.7699	0.5797	0.8889
	MLE-ADD	0.4876	0.7646	0.4911	0.7657	0.5870	**0.9205**
	MLE-REPLACE	0.4313	**0.7730** ⋆	0.4496	**0.8042**	0.5000	0.8961
	QFG	0.6049	0.6049	0.6859	0.6859	**0.6377**	0.6377
	AR	0.4586	0.5970	0.4975	0.6451	0.5217	0.6792
	ACO	0.4810	0.6774	0.5164	0.7238	0.5072	0.7143
Doc.-B.	SNIPPET	0.3054	0.3633	0.4375	0.5183	0.3623	0.4310
	NGRAMS-Site	0.2754	0.3689	0.3096	0.4127	0.3841	0.5146
	NGRAMS-MSN	0.2674	0.3075	0.3930	0.4506	0.3841	0.4417

Table 5: Results for all queries, • means statistical significance (at $p < 0.05$) over all other systems using two-tailed t-tests on the scores of individual assessors, ⋆ means significance over all systems other than *MLE-Dialoge-Add*.

Let us now discuss the results in more detail.

9 Discussion

We will approach the discussion from a range of different angles each representing different variables in our experimental setup. First of all, we will provide some general findings by comparing the results obtained for the different methods. We will then explore how the two different ways of segmenting our log files, i.e. according to sessions or according to "dialogues", affect the results. We will also investigate the results obtained from the three different user groups we sampled from, i.e. local users, Web users and the expert user. Following on from that we will discuss another important consideration, namely the question as to how much the suggestions derived for each method differ, in other words to find out which methods provide significantly different suggestions and which ones do not. Finally, before we discuss limitations of this study we will address the issue of data sparsity in relation to log data.

Method		Local Users		Web Users		Expert	
		$MP@3$	$MP@Max$	$MP@3$	$MP@Max$	$MP@3$	$MP@Max$
Log-Based	Session						
	MLE	0.6990	0.6990	0.7156	0.7156	0.8400	0.8400
	QFG	**0.7790**○	0.7790	**0.8444**	**0.8444**	0.7733	0.7733
	AR	0.6533	0.6533	0.6818	0.6818	0.7733	0.7733
	ACO	0.6952	0.6952	0.7114	0.7114	0.8000	0.8000
	Dialogue						
	MLE	0.7403	0.7820	0.7335	0.7740	0.8533	0.9014
	MLE-ADD	0.7390	0.7807	0.7225	0.7624	**0.8800**	**0.9296**
	MLE-REPLACE	0.6775	**0.8195**•	0.6666	0.8085	0.7600	0.9194
	QFG	0.7702	0.7702	0.7913	0.7913	0.8400	0.8400
	AR	0.6571	0.6571	0.6742	0.6742	0.7600	0.7600
	ACO	0.7067	0.7067	0.7362	0.7362	0.7200	0.7200
Doc.-B.	SNIPPET	0.3848	0.3953	0.5482	0.5629	0.4400	0.4521
	NGRAMS-Site	0.3632	0.3783	0.4008	0.4172	0.5067	0.5278
	NGRAMS-MSN	0.3302	0.3302	0.4793	0.4793	0.4933	0.4933

Table 6: Results for randomly sampled frequent queries,○ means statistical significance (at $p < 0.05$) over all other systems using two-tailed t-tests on the scores of individual assessors other than *QFG-Dialogue*, • means significance over all systems.

Method		Local Users		Web Users		Expert	
		$MP@3$	$MP@Max$	$MP@3$	$MP@Max$	$MP@3$	$MP@Max$
Log-Based	Session						
	MLE	0.3371	0.6321	0.3732	0.6940	0.3467	0.6500
	QFG	**0.4724**†	0.4724	0.5994	0.5994	0.4000	0.4000
	AR	0.4425	0.5029	0.5378	0.6129	0.4000	0.4545
	ACO	0.4590	0.5553	0.5378	0.6547	0.4133	0.5000
	Dialogue						
	MLE	0.2819	0.6820	0.3123	0.7521	0.3333	0.8065
	MLE-ADD	0.2717	**0.7028**‡	0.2934	0.7563	0.3200	**0.8276**
	MLE-REPLACE	0.2476	0.6878	0.2934	**0.8120**	0.2933	0.8148
	QFG	0.4679	0.4679	**0.6022**	0.6022	**0.4667**	0.4667
	AR	0.2851	0.4972	0.3424	0.5949	0.3200	0.5581
	ACO	0.2902	0.6218	0.3354	0.7160	0.3200	0.6857
Doc.-B.	SNIPPET	0.2794	0.3810	0.3711	0.5033	0.3733	0.5091
	NGRAMS-Site	0.2070	0.3610	0.2444	0.4241	0.3067	0.5349
	NGRAMS-MSN	0.2102	0.2765	0.3291	0.4316	0.3067	0.4035

Table 7: Results for randomly sampled queries from complete log,† means statistical significance (at $p < 0.01$) over all other systems using two-tailed t-tests on the scores of individual assessors other than *QFG-Dialogue* and *ACO*, ‡ means significance (at $p < 0.05$) over all systems other than *MLE-Dialogue-Replace*.

9.1 Method Comparison

We can derive a number of general observations from the results displayed in Tables 5, 6 and 7.

First of all, we find that log-based methods significantly outperform any of the non-log-based methods we applied. In other words, any of the methods that extract query suggestions from past queries generate more relevant query suggestions than either the snippets of top documents or different n-gram statistics extracted from document collections.

A second main observation, interpreting our measure $MP@Max$, is that maximum likelihood estimates applied to pairs of queries results in more relevant suggestions than any of the other approaches no matter whether or not the data is segmented according to sessions or dialogues. The caveat however is sparsity. A large number of queries – less frequent queries in particular – will have fewer than 3 suggestions (possibly none at all) proposed by MLE-based methods. This is clearly reflected by the huge gap between $MP@3$ and $MP@Max$ for these queries, most strikingly in Table 7. On the other hand, Query Flow Graphs are the best-performing method according to $MP@3$, with more consistent results across both measures. For frequent queries, in particular QFGs are performing well, in the case of rare queries QFGs might recommend unrelated queries whereas MLE will typically return nothing. Table 4 is an illustration of both the fact that QFGs can lead to suggestions where MLE fails to do so but also that some of these suggestions may be unrelated or not very useful.

When we focus on the log-based methods only, we find that the association rules approach performs less well than the other log-based methods, e.g. significantly worse than QFG. Part of the reason for this finding appears to be the fact that the AR method ignores the order in which two queries were submitted. This can sometimes be a problem. For example, as illustrated in Table 3, given the (top 10) query *"accommodation"*, the method *AR-Dialogue* suggests *"accomodation"*, a misspelling often seen as an initial query followed by the correct spelling. *MLE-Dialogue* and *QFG* on the other hand propose *"accommodation services"* and *"accommodation information"*, respectively. Misspelled queries can make up a large proportion of queries submitted on a local Web site, e.g. (Sutcliffe, White, and Kruschwitz, 2010). In fact, our sampled queries in Table 2 contain four queries that are misspelled.

Another word about MLE. As an alternative to MLE we considered *Pairwise Mutual Information (MI)* and *Chi square* (χ^2). However, these methods tend to favour rare query bigrams (Croft, Metzler, and Strohman, 2009). Therefore even for frequent queries, they promote those query pairs where the follow-up query is rather specific. For example, the MI approach applied to our sample query *"timetable"* yields *"timetable psychology"* instead of a more general relation such as *"exam timetable"*. This observation is in line with (Koren, Zhang, and Liu, 2008), who found MI to work poorly (for the same reason) in selecting facets and facet values in faceted search.

9.2 Query Log Segmentation

In Section 5 we motivated two different approaches of segmenting queries in a log file into what we consider the same search mission. The first approach is a session identification method, the second one is more fine-grained approach, and we refered to it as grouping queries that fall within the same "dialogue".

Looking at the results, we can see that dialogue-based query segmentation tends to perform better than session-based approaches when assessing the actually extracted suggestions (i.e. using $MP@Max$). We conclude that a session can be too coarse-grained for defining where a search mission starts and where it ends. Instead, we find that if we define the boundaries of a search mission by looking for explicit modifications of the initial query (addition or replacement

of terms as part of a dialogue), we can derive more relevant query suggestions. Again, one of the drawbacks is data sparsity. An approach that aims at breaking interaction sequences into even smaller units (by only considering query substitution or refinement sequences as related queries) only gives marginal further improvement.

9.3 Users

When we compare the judgements obtained from local users against general Web users we observe that there is a close correspondence for suggestions derived from local query logs. In addition to that, all log-based methods beat any of the three baselines when considering the full sample or frequent queries only (most of the differences are significant). All this indicates that these relations are in fact less domain-specific than originally anticipated and are perhaps applicable in general Web search and not just site-specific search. However, general Web users judge the baseline suggestions to be more relevant than local users which seems rather intuitive particularly for the MSN n-gram suggestions.

An interesting observation is that the domain expert judged far more suggestions proposed by the different methods as relevant than the average local user. Closer inspection however reveals that this is consistent throughout the different methods (the ranked orders of the methods' performance is almost identical to that of the local users). This might be explained by a broader domain knowledge the expert has as being responsible for the entire site. The expert judgements suggest that the Web site administrator would consider about nine in ten suggestion derived from MLE applied to dialogues to be relevant, but it also illustrates that for random queries there is on average just a single such suggestion.

Table 8 shows a slightly different perspective on the results presented in Tables 5, 6 and 7. Here we give a breakdown of what types of queries end up with good suggestions and which do not and get an indication of where disagreements between the user groups lie. We can see that on aggregate frequent queries result in better suggestions than average or less frequent queries. This is true for all user groups. We also observe, perhaps somehow surprisingly, that the expert is more likely to select *"Don't Know"* for the relevance of a suggestion than the other user groups and similarly is less likely to select *"Not relevant"*.

The findings regarding different user groups indicate that there is room for future investigations. At this point we want to present possible explanations that can be tested in future experiments. First of all, we observe that the general trend with all three sets of users is similar. The reluctance of the Web users (CrowdFlower recruits) to select *"Don't Know"* could be explained by their reluctance to show a lack of effort or conversely their eagerness to display a sense of effort expended. The expert's reluctance to select *"Not relevant"* could be explained by his more technical background and insight. He is aware that many of these suggestions are derived from log data and although seemingly unrelated, chooses *"Don't Know"* where there is a degree of plausibility.

9.4 Suggestion Analysis

Obviously, some of the methods we introduced are highly correlated (e.g. the different flavours of MLE) whereas others will present significantly different query suggestions (e.g. any log-based method compared to any document-based approach). To get a more complete picture of the different methods and their performance we conducted correlation tests that compare the suggestions derived for different methods. We applied a pairwise comparison between any pair of methods using three different metrics each of which reflecting a slightly different correlation

Query Sample	Subjects	Judgements		
		Relevant	Not Relevant	Don't Know
All	Expert	0.6583	0.1971	0.1446
	Local	0.5832	0.3083	0.1085
	Web	0.6494	0.3037	0.0469
Frequent	Expert	0.7461	0.1338	0.1201
	Local	0.6477	0.2983	0.0540
	Web	0.6880	0.2697	0.0423
Random	Expert	0.5407	0.2759	0.1834
	Local	0.4998	0.3239	0.1763
	Web	0.6070	0.3420	0.0510

Table 8: Query sample performance per subject type – a collation of the three detailed query sample tables

between suggestions, two of them treating the suggestions as a bag of words and one of them as an ordered list. Let us assume, that $S_{qi} = \{s_{qi1}, s_{qi2}, s_{qi3}\}$, is the list of suggestions for query q, using method i. The metrics we use are then defined as follows:

- *Dice's coefficient* is a *symmetric* measure that compares two sets of suggestions as follows:

$$d_{ij} = \frac{2 * |S_i \cap S_j|}{|S_i| + |S_j|} \tag{10}$$

  where $|S_i| = \sum_{q=1}^{46} |S_{qi}|$ and does not include blank suggestions

- *Percent Overlap* is a similar metric we define but which is *non-symmetric* and motivated analogously to $MP@Max$, i.e., this metric reflects the number of suggestions generated by the method. Percent overlap of method i with method j is defined as follows:

$$O_{ij} = \frac{\sum_{q=1,k=1}^{q=46,k=3} (o_{qk})}{\sum_{q=1}^{46} |S_{qi}|} \tag{11}$$

  where

$$o_{qk} = \begin{cases} 1 & \text{if } s_{qik} \text{ exists and } s_{qik} \in S_{qj} \\ 0 & \text{Otherwise} \end{cases} \tag{12}$$

- *Kendall's Tau* τ is a symmetric rank correlation coefficient that compares two ordered lists and we apply it as suggested in (Baeza-Yates and Ribeiro-Neto, 2010)[7]

Table 9 presents the results of applying each of the three metrics in a pairwise comparison of all the different methods investigated in this study. We can derive a number of observations from the suggestion analysis.

First of all, the figures demonstrate that there is almost no overlap between any of the suggestions derived from log-based methods when compared to the document-based methods as

[7]We consider a number of special cases, namely if both lists are empty, we have total agreement ($\tau = 1$), else if one list only is empty, we have total disagreement ($\tau = 0$), else if each list contains one same suggestion, we have total agreement ($\tau = 1$), otherwise we calculate τ as suggested in (Baeza-Yates and Ribeiro-Neto, 2010).

		Method	**Log-Based**																												**Document-Based**								
			Session												**Dialogue**																								
			MLE			QFG			AR			ACO			MLE-D			MLE-DA			MLE-DR			QFG-D			AR-D			ACO-D			SNIPPET			NGRAM-S			NGRAM-M		
			%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d	%	τ	d
Log-Based	Session	MLE	1.00	1.00	1.00	0.41	0.07	0.35	0.52	0.14	0.47	0.90	0.67	0.82	0.49	0.31	0.52	0.50	0.24	0.53	0.39	0.31	0.44	0.38	0.08	0.32	0.25	0.07	0.25	0.40	0.20	0.41	0.02	-0.06	0.02	0.05	-0.03	0.05	0.00	-0.06	0.00
		QFG	0.30	-	-	1.00	1.00	1.00	0.28	-0.08	0.28	0.45	0.06	0.47	0.31	0.12	0.38	0.32	0.10	0.39	0.27	0.08	0.34	0.51	0.26	0.51	0.17	-0.08	0.19	0.25	-0.01	0.30	0.01	-0.16	0.01	0.02	-0.16	0.02	0.01	-0.18	0.01
		AR	0.42	-	-	0.29	-	-	1.00	1.00	1.00	0.47	0.08	0.48	0.22	-0.01	0.26	0.23	-0.05	0.28	0.17	-0.02	0.21	0.19	-0.07	0.18	0.27	0.02	0.30	0.18	0.01	0.20	0.02	-0.14	0.02	0.06	-0.10	0.07	0.02	-0.12	0.02
		ACO	0.74	-	-	0.50	-	-	0.49	-	-	1.00	1.00	1.00	0.36	0.12	0.42	0.35	0.11	0.41	0.30	0.12	0.38	0.34	0.00	0.33	0.22	-0.05	0.24	0.32	0.09	0.36	0.02	-0.13	0.02	0.04	-0.11	0.04	0.00	-0.12	0.00
	Dialogue	MLE-D	0.56	-	-	0.48	-	-	0.31	-	-	0.50	-	-	1.00	1.00	1.00	0.81	0.71	0.82	0.73	0.64	0.79	0.66	0.24	0.52	0.31	0.22	0.29	0.53	0.35	0.51	0.01	0.01	0.01	0.03	0.04	0.03	0.00	0.00	0.00
		MLE-DA	0.58	-	-	0.50	-	-	0.34	-	-	0.50	-	-	0.83	-	-	1.00	1.00	1.00	0.57	0.49	0.61	0.59	0.19	0.46	0.30	0.24	0.27	0.48	0.33	0.45	0.01	0.01	0.01	0.03	0.04	0.03	0.00	0.00	0.00
		MLE-DR	0.52	-	-	0.48	-	-	0.29	-	-	0.49	-	-	0.86	-	-	0.65	-	-	1.00	1.00	1.00	0.68	0.18	0.48	0.34	0.22	0.28	0.55	0.31	0.48	0.01	0.02	0.01	0.04	0.06	0.03	0.00	0.01	0.00
		QFG-D	0.28	-	-	0.51	-	-	0.17	-	-	0.31	-	-	0.43	-	-	0.38	-	-	0.38	-	-	1.00	1.00	1.00	0.24	-0.03	0.27	0.30	0.11	0.35	0.01	-0.16	0.01	0.01	-0.15	0.02	0.01	-0.17	0.01
		AR-D	0.25	-	-	0.22	-	-	0.33	-	-	0.26	-	-	0.26	-	-	0.25	-	-	0.25	-	-	0.31	-	-	1.00	1.00	1.00	0.32	0.25	0.33	0.01	-0.01	0.01	0.02	0.04	0.02	0.01	-0.02	0.01
		ACO-D	0.42	-	-	0.36	-	-	0.23	-	-	0.41	-	-	0.49	-	-	0.43	-	-	0.43	-	-	0.42	-	-	0.35	-	-	1.00	1.00	1.00	0.01	0.01	0.01	0.02	0.03	0.02	0.02	0.00	0.02
Doc-Based		SNIPPET	0.02	-	-	0.01	-	-	0.02	-	-	0.02	-	-	0.01	-	-	0.01	-	-	0.01	-	-	0.01	-	-	0.01	-	-	0.01	-	-	1.00	1.00	1.00	0.01	-0.02	0.01	0.00	-0.06	0.00
		NGRAM-S	0.50	-	-	0.03	-	-	0.08	-	-	0.05	-	-	0.03	-	-	0.03	-	-	0.03	-	-	0.02	-	-	0.02	-	-	0.02	-	-	0.01	-	-	1.00	1.00	1.00	0.01	0.00	0.09
		NGRAM-M	0.00	-	-	0.01	-	-	0.02	-	-	0.00	-	-	0.00	-	-	0.00	-	-	0.00	-	-	0.01	-	-	0.01	-	-	0.02	-	-	0.00	-	-	0.08	-	-	1.00	1.00	1.00

Table 9: Suggestion analysis – Percent overlap (refered to as %), Kendall’s Tau agreement and Dice’s Coefficient (”-” reflects the symmetric nature of the metric and declutters the table).

can be seen from the values for *Percent overlap* and *Dice* which are zero or very close to zero in all pairwise comparisons. In addition to that we also observe that all three baseline methods tend to produce distinct suggestions. We conclude that the suggestions derived from either log-based or document-based approaches are complementary (in those cases where they are judged relevant by the users). This is in line with findings from similar experiments conducted on digital libraries (Kruschwitz et al., 2011).

Secondly, there are some clusters of methods that appear to produce very similar suggestions. This is true for the different variants of dialogue-based MLE which explains the very similar judgements obtained for methods *MLE-Dialogue*, *MLE-Dialogue-Add* and *MLE-Dialogue-Replace* in Tables 5, 6 and 7. This reiterates the finding that trying to break search missions into smaller units than "dialogues" does not offer substantial improvements. Another notable fact is the close correspondence between session-based MLE and ACO. This can be explained by the fact that ACO is structurally similar to MLE, actually ACO without evaporation is almost identical to MLE. However, the two variants of ACO and MLE that use segmentation based on dialogue rather than session result in more dissimilar suggestions. Both these observations are in line with the examples in Table 3.

A final observation from the suggestion analysis is that the different variants of dialogue-based MLE produce similar suggestions to QFG but in a different ranking order (as indicated by high *Percent overlap* but very low corresponding τ values).

Regarding the different metrics we applied we also observe that overall *Percent overlap* and *Dice* tend to show close resemblance which is not surprising given the similarity in calculating the measures.

9.5 Discussion of Data Sparsity

A very common problem in data-driven approaches is data sparsity (already reflected by the example in Table 4). The results of our study clearly support that this is a general problem for a number of log-based methods. In particular, suggestions derived for average queries (rather than frequent queries) suffer a lot from sparsity. Table 7 illustrates this by the large margins between the different metrics $MP@3$ and $MP@Max$. For example, comparing the results we get for method *MLE-ADD* we see that the suggestions derived from the log files are generally considered relevant by any of the user groups (reflected by $MP@Max$) but very few such suggestions are actually generated (see results for $MP@3$).

Naturally, any log-based method can only be as good as the data is it applied to. Therefore we explored how the sparsity issue is reflected by the raw log files. Using the sample queries, let us look at how many times a query has a modification and how different the modifications are. Table 10 presents this information for five frequent queries and five of the randomly sampled queries. The table includes information about the total number of times the query has been submitted, the number of times this query was follwed by any modification (addition of terms or replacement) and the number of unique modifications. These last two figures give an idea about the type/token ratio of the query modifications. We also include information about the overall average, i.e. not just the sampled queries. No matter whether we include those queries that never get modified at all or whether we exclude those in the calculation (see figures in the parentheses in Table 10), we observe that on average the log data concerning query modifications is sparse.

Judging by the results obtained from local users and the domain expert, we conclude that *all* log-based methods perform better on frequent queries than queries sampled from the entire log. A conclusion for practical applications could be that only frequent (and possibly medium-frequent) queries are at all considered for generating query modification suggestions in a live

Query	**Total in Logs**	**Modifications**	**Unique Modifications**
moodle	74,240	8,291	1,137
library	33,419	4,625	1,016
timetable	21,801	4,610	1,512
cmr	10,980	1,562	536
enrol	8,551	1,905	608
phd	2,298	957	559
frontrunners	1,931	249	157
staff research	6	3	3
viviene jamescook	1	0	0
report incident	1	1	1
Overall average (with modification)	*5.55*	*2.02* (*3.33*)	*1.41* (*2.33*)

Table 10: Sampled queries: some general data.

system. Other approaches do exist to address the issue of long-tail queries, for example one could abstract from the actual queries and use templates instead (Szpektor, Gionis, and Maarek, 2011).

9.6 Limitations

It is difficult to make direct comparisons between our findings and those of previous studies. To the best of our knowledge, there has been no comparable study that assesses query suggestions derived for site search or intranet search. For Web documents, Sanderson and Croft found that 67% of terms derived from a term hierarchy (constructed using the retrieved documents) were judged "interesting" (Sanderson and Croft, 1999). (Boldi et al., 2009) used query flow graphs on Web logs and their best methods produced 58% of suggestions that are either "useful" or "somewhat useful". Unlike in our study, however, assessments were performed for five rather than three recommendations per query.

Results of other log-analysis studies are more difficult to compare as the experimental setup differs. For example, (Fonseca et al., 2003) found that automatically generated query suggestions based on previously submitted Web queries can be "correct" in more than 90%, (Baeza-Yates, Hurtado, and Mendoza, 2004) report similar results. However, both their evaluations did not involve actual users but people in the research groups who had to assess whether they believed that the suggestion could be *interesting* for users who formulated the original query, a very different scenario.

An obvious limitation of such a study is that the results are based on data from a single Web site and the findings may or may not be transferable to other document collections. This is also true for any comparison of the results with studies conducted on *Web* logs. Site logs may have very different characteristics. While Web queries tend to be around 2.35 words long on average, e.g. (Silverstein, Henzinger, and Marais, 1998; Jansen, Bateman, and Saracevic, 1998; Beitzel et al., 2004; Beitzel et al., 2007), our queries are shorter, on average 1.81 query terms which is consistent with a study we conducted on a 2001 log file within the same domain (average query length of 1.72) (Kruschwitz, 2003) and supported by results reported for intranets, e.g. (Stenmark and Jadaan, 2006). The same appears to be true for sessions. Our logs contain on average 1.53 queries per session, compared to 2.02 queries per session on average on an *AltaVista* log (Silverstein, Henzinger, and Marais, 1998), 2.8 for an *Excite* log (Jansen, Bateman,

and Saracevic, 1998) and 2.31 as the average number of queries per session submitted to the meta-search engine *Dogpile* (using a session definition similar to ours) (Jansen et al., 2007). Interestingly, another study of a local Web site has also come to the conclusion that sessions are shorter than general Web search sessions: on average 1.73 queries were submitted per session to the Utah government Web site (Chau, Fang, and Sheng, 2005).

Despite the limitations, we see two main contributions that our study makes. First of all, it demonstrates how to systematically evaluate query suggestions (prior to assessing them in a live system). Secondly, our results could serve as a baseline for future studies on different Web sites. The major bottleneck in conducting research into query logs is the difficulty in getting hold of *realistic* and *large-scale* log data which is also the reason why it is nearly impossible to conduct and report studies on a selection of large-scale logs collected on different sites.

10 Conclusions

We presented a systematic study on methods for query modification suggestions on a local Web site. We explored a range of log-based methods and several baselines that do not exploit query logs. Furthermore we investigated a second variable when exploiting logs, the way how sessions can be segmented into sequences of queries related to the same search mission. We will now draw conclusions with reference to our original research questions set in Section 3.

1. *Are log files of an interactive local search system an appropriate resource to derive query modification suggestions for site-specific search?*

   We found that all our log-based approaches, adaptive or non-adaptive, significantly outperform non-log-based baselines. Regarding the methods, we observe a trade-off between highly accurate but sparse methods (MLE) and methods that are generating more suggestions even for less frequent queries (QFG). The experimental results confirm that logs represent a very valuable resource to automatically acquire feedback terms for guided search.

   We observed that log analysis techniques which group queries according to sessions tend to result in relevant query suggestions. We also conclude that more fine-grained groupings (in this case according to interaction dialogues) offer the potential to derive even better suggestions, in particular for frequently submitted queries. As discussed previously, automatically identifying the boundaries of sessions is a difficult task, and one of the reasons is that a session can easily consist of a number of search goals. Our results demonstrate that recording such information in the log file has the potential of substantially improving search suggestions.

2. *Is the relevance of such feedback terms perceived differently by searchers on the local Web site compared to the general Web population?*

   We observe that there is a close correspondence between the assessments provided by local users and Web users for suggestions derived from local query logs. In addition to that, the log-based methods beat any of the non-log-based baselines. This indicates that locally collected log files might actually be more transferable to generic Web search than intuitively expected.

11 Future Work

There are a number of future directions worth exploring. Validating the findings on different Web sites is one of the obvious tasks to be conducted.

An interesting area of study is the question as to how far back we want to go in the interaction history when extracting relations. Here we used a log file recording three years of interaction. There are however a number of temporal queries that are either popular only for a short period of time or which are submitted only at certain times of the year, and queries which ask for different refinement suggestions at different times of the year. For example, *"timetable"* is a frequently submitted query but in autumn this often refers to the teaching timetable whereas in spring it is more likely that students search for the exam timetable.

Another area that we did not explore is whether queries should perhaps not just be related if they follow each other immediately but also if there are other dialogue steps in between, so an initial query might be treated as related to any follow-up query that happens within the same dialogue.

Future investigations will need to go beyond the assessment of simple term relations and apply the methods in controlled longitudinal studies. For example, to find out if the suggestions allow the users to get to relevant information more easily, whether the application of the log analysis methods will reinforce certain selections, and whether the search framework will result in better overall user satisfaction. As one of our next steps we are going to conduct extensive A/B testing in a live environment (Kohavi, Henne, and Sommerfield, 2007).

Finally, a promising direction is to construct some domain knowledge not simply based on query logs but start with a model that has been bootstrapped from the document collection and then adapt that model using query logs (Adeyanju et al., 2012). Such a model has a number of advantages, one of them is that it will suffer less from data sparsity that we had observed in particular with the less common queries.

Acknowledgements

This research is part of the AutoAdapt research project. AutoAdapt is funded by EPSRC grants EP/F035357/1 and EP/F035705/1. We would also like to thank the anonymous reviewers for very constructive comments that allowed us to substantially extend the analysis of the results in particular.

References

Adeyanju, I., D. Song, M-D. Albakour, U. Kruschwitz, M. Fasli, and A. De Roeck. 2012. Adaptation of the Concept Hierarchy model with search logs for query recommendation on Intranets. In *Proceedings of 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 5–14, Portland (Oregon).

Al Hasan, M., N. Parikh, G. Singh, and N. Sundaresan. 2011. Query Suggestion for E-Commerce Sites. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM'11)*, pages 765–774, Hong Kong.

Albakour, M-D. 2012. *Adaptive Domain Modelling for Information Retrieval*. Ph.D. thesis, University of Essex.

Albakour, M-D., U. Kruschwitz, and S. Lucas. 2010. Sentence-level attachment prediction. In *Proceedings of the 1st Information Retrieval Facility Conference*, volume 6107 of *Lecture Notes in Computer Science*, pages 6–19, Vienna. Springer.

Albakour, M-D., U. Kruschwitz, N. Nanas, D. Song, M. Fasli, and A. De Roeck. 2011a. Exploring Ant Colony Optimisation for Adaptive Interactive Search. In *Proceedings of the 3rd International Conference on the Theory of Information Retrieval (ICTIR)*, Lecture Notes in Computer Science, pages 213–224, Bertinoro. Springer.

Albakour, M-D., N. Nanas, U. Kruschwitz, M. Fasli, Y. Kim, D. Song, and A. De Roeck. 2011b. AutoEval: An Evaluation Methodology for Evaluating Query Suggestions Using Query Logs. In *Proceedings of the 33rd European Conference on Information Retrieval (ECIR'11)*, volume 6611 of *Lecture Notes in Computer Science*, pages 605–610, Dublin. Springer.

Anick, P. 2003. Using Terminological Feedback for Web Search Refinement - A Log-based Study. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 88–95, Toronto, Canada.

Baeza-Yates, R., C. Hurtado, and M. Mendoza. 2004. Query recommendation using query logs in search engines. In *Current Trends in Database Technology - EDBT 2004 Workshops, Workshop on Clustering Information over the Web, Heraklion, Crete, Greece, March 14-18, 2004, Revised Selected Papers*, Lecture Notes in Computer Science 3268. Springer, pages 588–596.

Baeza-Yates, R. and B. Ribeiro-Neto, editors. 2010. *Modern Information Retrieval*. Addison-Wesley, 2nd edition.

Baeza-Yates, R. and F. Saint-Jean. 2003. A Three Level Search Engine Index Based in Query Log Distribution. In *Proceedings of the 10th International Symposium on String Processing and Information Retrieval (SPIRE)*, Lecture Notes in Computer Science 2857, pages 56–65, Manaus, Brazil.

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

Beitzel, S. M., E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. 2007. Temporal Analysis of a Very Large Topically Categorized Web Query Log. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(2):166–178, January.

Beitzel, S. M., E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. 2004. Hourly Analysis of a Very Large Topically Categorized Web Query Log. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–328, Sheffield.

Belkin, N. J. 2008. Some(what) grand challenges for information retrieval. *SIGIR Forum*, 42(1):47–54.

Ben-Yitzhak, O., N. Golbandi, N. Nadav, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev. 2008. Beyond basic faceted search. In *Proceedings of the First ACM International Conference on Web Search and Data Mining (WSDM'08)*, pages 33–44, Palo Alto.

Boldi, P., F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. 2008. The query-flow graph: model and applications. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM'08)*, pages 609–618.

Boldi, P., F. Bonchi, C. Castillo, D. Donato, and S. Vigna. 2009. Query suggestions using query-flow graphs. In *Proceedings of the 2009 workshop on Web Search Click Data (WSCD'09)*, pages 56–63.

Callison-Burch, C. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 286–295. Association for Computational Linguistics.

Chau, M., X. Fang, and O. R. L. Sheng. 2005. Analysis of the Query Logs of a Web Site Search Engine. *Journal of the American Society for Information Science and Technology (JASIST)*, 56(13):1363–1376, November.

Clark, M., Y. Kim, U. Kruschwitz, D. Song, M-D. Albakour, S. Dignum, U. Cervino Beresi, M. Fasli, and A. De Roeck. 2012. Automatically Structuring Domain Knowledge from Text: an Overview of Current Research. *Information Processing and Management*, 48(3):552–568.

Craswell, N. and M. Szummer. 2007. Random Walks on the Click Graph. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 239–246, Amsterdam.

Croft, B., D. Metzler, and T. Strohman. 2009. *Search Engines: Information Retrieval in Practice*. Pearson.

Deng, H., I. King, and M. Lyu. 2009. Entropy-biased models for query representation on the click graph. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 339–346, Boston.

Di Caro, G. and M. Dorigo. 1998. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365.

Dignum, S., U. Kruschwitz, M. Fasli, Y. Kim, D. Song, U. Cervino, and A. De Roeck. 2010. Incorporating Seasonality into Search Suggestions Derived from Intranet Query Logs. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence (WI'10)*, pages 425–430, Toronto.

Dorigo, M., M. Birattari, and T. Stutzle. 2006. Ant colony optimization. *IEEE Intelligent Systems*, 1:28–39.

Dumais, S., T. Joachims, K. Bharat, and A. Weigend. 2003. SIGIR 2003 workshop report: implicit measures of user interests and preferences. *SIGIR Forum*, 37(2):50–54.

Efthimiadis, E. N. 1996. Query Expansion. In M. E. Williams, editor, *Annual Review of Information Systems and Technology (ARIST)*, volume 31. Information Today, pages 121–187.

Fonseca, B. M., P. B. Golgher, E. S. de Moura, B. Pôssas, and N. Ziviani. 2004. Discovering search engine related queries using association rules. *Journal of Web Engineering*, 2(4):215–227.

Fonseca, B. M., P. B. Golgher, E. S. de Moura, and N. Ziviani. 2003. Using association rules to discover search engines related queries. In *Proceedings of the First Latin American Web Congress*, pages 66–71.

Gayo-Avello, D. 2009. A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, 179:1822–1843, May.

Göker, A. and D. He. 2000. Analysing web search logs to determine session boundaries for user-oriented learning. In *AH '00: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 319–322. Springer.

Hawking, D. 2010. Enterprise Search. In R. Baeza-Yates and B. Ribeiro-Neto, editors, *Modern Information Retrieval*. Addison-Wesley, 2nd edition, pages 645–686.

Huang, J. and E. N. Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceeding of CIKM'09*. ACM.

Jansen, B. J., J. Bateman, and T. Saracevic. 1998. Real life information retrieval: A study of user queries on the web. *SIGIR Forum*, 32(1):5–17.

Jansen, B. J., A. Spink, C. Blakely, and S. Koshman. 2007. Defining a session on Web search engines. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(6):862–871, April.

Jansen, B. J., A. Spink, and S. Koshman. 2007. Web Server Interaction with the Dogpile.com Metasearch Engine. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(5):744–755, March.

Jansen, B. J., A. Spink, and T. Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227.

Jansen, J., A. Spink, and I. Taksa, editors. 2008. *Handbook of Research on Web Log Analysis*. IGI.

Joachims, T., L. Granka, B. Pan, H. Hembrooke, and G. Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, Salvador, Brazil.

Joachims, T. and F. Radlinski. 2007. Search engines that learn from implicit feedback. *IEEE Computer*, 40(8):34–40.

Jones, R. and K. L. Klinkner. 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM'08)*, pages 699–708.

Jones, R., B. Rey, O. Madani, and W. Greiner. 2006. Generating Query Substitutions. In *Proceedings of the 15th International World Wide Web Conference (WWW'06)*, pages 387–396, Edinburgh.

Justeson, J. S. and S. M. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Kelly, D., K. Gyllstrom, and E. W. Bailey. 2009. A comparison of query and term suggestion features for interactive searching. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 371–378, Boston.

Kohavi, R., R. M. Henne, and D. Sommerfield. 2007. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 959–967, New York, NY, USA. ACM.

Koren, J., Y. Zhang, and X. Liu. 2008. Personalized Interactive Faceted Search. In *Proceeding of the 17th International World Wide Web Conference (WWW'08)*, pages 477–486, Beijing.

Kraft, R. and J. Zien. 2004. Mining Anchor Text for Query Refinement. In *Proceedings of the 13th International World Wide Web Conference (WWW2004)*, pages 666–674, New York.

Kruschwitz, U. 2003. An Adaptable Search System for Collections of Partially Structured Documents. *IEEE Intelligent Systems*, 18(4):44–52, July/August.

Kruschwitz, U. 2005. *Intelligent Document Retrieval: Exploiting Markup Structure*, volume 17 of *The Information Retrieval Series*. Springer.

Kruschwitz, U. and H. Al-Bakour. 2005. Users Want More Sophisticated Search Assistants - Results of a Task-Based Evaluation. *Journal of the American Society for Information Science and Technology (JASIST)*, 56(13):1377–1393, November.

Kruschwitz, U., M-D. Albakour, J. Niu, J. Leveling, N. Nanas, Y. Kim, D. Song, M. Fasli, and A. De Roeck. 2011. Moving towards Adaptive Search in Digital Libraries. In *Advanced Language Technologies for Digital Libraries*, volume 6699 of *Lecture Notes in Computer Science*. Springer, pages 41–60.

Kruschwitz, U., N. Webb, and R. F. E. Sutcliffe. 2008. Query Log Analysis for Adaptive Dialogue-Driven Search. In J. Jansen, A. Spink, and I. Taksa, editors, *Handbook of Research on Web Log Analysis*. IGI, Hershey, PA, pages 389–416.

Liu, C., J. Gwizdka, J. Liu, T. Xu, and N. J. Belkin. 2010. Analysis and evaluation of query reformulations in different task types. In *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem*.

Lungley, D. and U. Kruschwitz. 2009. Automatically maintained domain knowledge: Initial findings. In *Proceedings of the 31st European Conference on Information Retrieval (ECIR'09)*, pages 739–743, Toulouse.

Manning, C. D. and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Marchionini, G. 2008. Human-information interaction research and development. *Library and Information Science Research*, 30(3):165–174.

Markey, K. 2007. Twenty-five years of end-user searching, Part 1: Research findings. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(8):1071–1081, June.

Martens, D., M. De Backer, J. Vanthienen, M. Snoeck, and B. Baesens. 2007. Classification with Ant Colony Optimization. *IEEE Transactions on Evolutionary Computation*, 11:651–665.

Radlinski, F., M. Kurup, and T. Joachims. 2008. How does clickthrough data reflect retrieval quality? In *Proceeding of CIKM'08*. ACM.

Ruthven, I. 2003. Re-examining the potential effectiveness of interactive query expansion. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 213–220, Toronto, Canada.

Ruthven, I. 2008. Interactive information retrieval. *Annual Review of Information Science and Technology (ARIST)*, 42:43–92.

Sanderson, M. and B. Croft. 1999. Deriving concept hierarchies from text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–213, Berkeley, CA.

Sherman, C. 2008. Why Enterprise Search will never be Google-y. *Enterprise Search Sourcebook*, pages 12–13.

Silverstein, C., M. Henzinger, and H. Marais. 1998. Analysis of a Very Large AltaVista Query Log. Digital SRC Technical Note 1998-014.

Silvestri, F. 2010. *Mining Query Logs: Turning Search Usage Data into Knowledge*, volume 4 of *Foundations and Trends in Information Retrieval*. Now Publisher.

Smyth, B., P. Briggs, M. Coyle, and M. O'Mahony. 2009. Google Shared. A Case-Study in Social Search. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization (UMAP)*, pages 283–294. Springer.

Snow, R., B. O'Connor, D. Jurafsky, and A. Y. Ng. 2008. Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics.

Socha, K., M. Sampels, and M. Manfrin. 2003. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In *Applications of Evolutionary Computing*, volume 2611 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pages 334–345.

Stenmark, D. and T. Jadaan. 2006. Intranet Users' Information-Seeking Behaviour: A Longitudinal Study of Search Engine Logs. In *Proceedings of ASIS&T*, Austin, TX.

Sutcliffe, R., K. White, and U. Kruschwitz. 2010. Named Entity Recognition in an Intranet Query Log. In *Proceedings of the LREC Workshop on Semitic Languages*, pages 43–49, Valletta, Malta.

Szpektor, I., A. Gionis, and Y. Maarek. 2011. Improving recommendation for long-tail queries via templates. In *Proceedings of the Twenteeth International World Wide Web Conference (WWW'11)*, pages 47–56, Hyderabad, India.

Tunkelang, D. J. 2009. *Faceted search*. Morgan & Claypool Publishers.

Wang, K., C. Thrasher, E. Viegas, X. Li, and B. Hsu. 2010. An overview of Microsoft web N-gram corpus and applications. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, HLT-DEMO '10, pages 45–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, P., M. W. Berry, and Y. Yang. 2003. Mining Longitudinal Web Queries: Trends and Patterns. *Journal of the American Society for Information Science and Technology (JASIST)*, 54(8):743–758, June.

White, M. 2007. *Making Search Work: Implementing Web, Intranet and Enterprise Search*. Facet Publishing.

White, R. W., M. Bilenko, and S. Cucerzan. 2007. Studying the Use of Popular Destinations to Enhance Web Search Interaction. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 159–166, Amsterdam.

White, R. W. and I. Ruthven. 2006. A Study of Interface Support Mechanisms for Interactive Information Retrieval. *Journal of the American Society for Information Science and Technology (JASIST)*, 57(7):933–948.

Widdows, D. and B. Dorow. 2002. A Graph Model for Unsupervised Lexical Acquisition and Automatic Word-Sense Disambiguation. In *Proceedings of the 19th Conference on Computational Linguistics (COLING)*, pages 1093–1099, Taipei, Taiwan.

Yin, Z., M. Shokouhi, and N. Craswell. 2009. Query expansion using external evidence. In *Proceedings of the 31st European Conference on Information Retrieval (ECIR'09)*, pages 362–374, Toulouse.

Yuan, X. and N. J. Belkin. 2007. Supporting multiple information-seeking strategies in a single system framework. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 247–254, Amsterdam.

Zamir, O. E., J. L. Korn, A. B. Fikes, and S. R. Lawrence. 2005. Personalization of Placed Content Ordering in Search Results. World Patent WO/2006/017364.