A survey on session detection methods in query logs and a proposal for future evaluation

Daniel Gayo-Avello[1]

Department of Computer Science, University of Oviedo, SPAIN

Abstract

Search engine logs provide a highly detailed insight of users' interactions. Hence, they are both extremely useful and sensitive. The datasets publicly available to scholars are, unfortunately, too few, too dated and too small. There are few because search engine companies are reluctant to release such data; they are dated because they were collected in late 1990s or early 2000s; and they are small because they comprise data for at most one day and just a few hundreds of thousands of users. Even worse, the large query log disclosed by AOL in 2006 caused more harm than good because of a big privacy flaw. In this paper the author provides an overall view of the possible applications of query logs, the privacy concerns researchers must face when working on such datasets, and several ways in which query logs can be easily sanitized. One of such measures consists of segmenting the logs into short topical sessions. Therefore, the author offers a comprehensive survey of session detection methods, as well as a thorough description of a new evaluation framework with performance results for each of the different methods. Additionally, a new, simple, but outperforming session detection method is proposed. It is a heuristic-based technique which works on the basis of a geometric interpretation of both the time gap between queries and the similarity between them in order to flag a topic shift.

Keywords: Web searching; search engine; query log; topical session; session detection

1. Introduction

Web Search companies keep log files detailing interaction of users with the search engine. The information typically recorded in these query logs includes a unique identifier for the user or the session, the query string, a timestamp and, occasionally, the results page number and the URLs clicked (if any) for each query. The analysis of such logs can provide an insight about searching behavior on the Web which is not only of interest for search engine companies but it notes the distinct features that differentiate Web Information Retrieval from classical IR.

The first in-depth studies on query logs date back to the late 1990s (e.g. [27, 33, 68, 69]). Such studies provided important details about Web searchers' behavior (e.g. query length, number of visited results, etc). Nevertheless, query logs can not only be analyzed to understand users' activities but also mined to develop novel search-related applications such as query suggestion (e.g. [6, 75, 77]) or re-ranking of search results (e.g. [36, 37]), among others.

Nonetheless, the resources available to scholars working outside search engine companies are scarce, dated and small. Researchers typically rely on a few publicly available query logs released by Excite [27, 33, 68, 69], AlltheWeb [70] and AltaVista [31] from 1997 to 2002. These files contain data for at most one day, each one comprising less than 3 million queries from just a few hundreds of thousands of users.

[1] Correspondence to: Daniel Gayo-Avello, Department of Computer Science (University of Oviedo) – Edificio de Ciencias, C/Calvo Sotelo s/n 33007 Oviedo (SPAIN), dani@uniovi.es

A much larger and recent dataset was announced by Microsoft Research on January 2006 [46]; however, this log –containing 15 million queries sampled over one month–was only disclosed to the few research groups awarded and, thus, it is not freely available for the academic community. Later, on August 2006, AOL publicly released a query log containing more than 30 million queries sampled over three months from over 650,000 users [56]. This data surpassed that provided by Microsoft, doubling the number of queries and covering a time span much longer.

The data disclosed by AOL had, however, an important flaw: it employed unique user IDs which could be used to group the queries by user across all the records. Because many users typically issue queries including sensitive information (e.g. personal names, addresses, or social security numbers) it is possible to analyze the data in order to identify individuals. A few users were eventually identified and one case exposed by the media [5]. The subsequent scandal led to the withdrawal of the data and put into question the ethics of researching on such query log. Several scholars were contacted by the media in order to elicit their opinion on the matter [3, 22]. The main conclusion one can achieves from such consultations is that any research not aiming the identification of actual people could be judged as ethically acceptable.

Thus, this paper addresses the problem of post-processing big query logs in order to dispel privacy concerns while preserving most of their usefulness for academic research. As it will be later discussed, short topical sessions can be a feasible way of segmenting such logs and, hence, this study deals with session boundary detection methods and their evaluation.

The paper is organized as follows. First of all, an extensive literature review is provided. It deals with query log analysis and mining; privacy issues and anonymization; definitions of searching episode and search session; and methods for session boundary detection. Then, the research questions are stated and a new sessionization technique is proposed. After that, the experimental framework in which this study was conducted is described: the used query logs, the size and nature of the samples, the elaboration of the ground truth files and the nature of the measures of evaluation. Afterwards, results obtained with each of the different sessionization methods are discussed along with the implications of the study.

2. Literature review

2.1. Query log analysis

The first in-depth study on a transaction log from a commercial search engine was conducted by Jansen *et al.* [33]. They worked on a query log collected by Excite on 9th March 1997 and containing about 51,000 entries from approximately 18,000 different users; this log is usually referred as *"Excite 1997 small"*.

During the following years Excite continued providing data and, thus, Spink *et al.* [68] described the so called *"Excite 1997 large"* log: around 1 million queries issued by over 210,000 different users on 16th September 1997; Jansen and Spink [27]; and Wolfram *et al.* [79] described the *"Excite 1999"* log consisting of a sample of 1 million queries from over 200,000 users collected on 20th December 1999; and Spink *et al.* [69] portrayed the *"Excite 2001"* log which contains over 1 million queries submitted by more than 250,000 users to the Excite search engine on 5th January 2001.

Logs from FAST's AlltheWeb.com and AltaVista were obtained and analyzed by Spink *et al.* [70] and Jansen, Spink and Pedersen [31], respectively; these logs contain 1 and 3 million records issued by 150,000 and 370,000 users on 6th February 2001 and 8th September 2002.

These studies have provided information about Web searchers' behavior such as the length of their queries (e.g. [31, 32, 33, 64, 65, 68]); the number of result pages they view (e.g. [28, 29, 30, 32, 78, 79]); the number of results they visit (e.g. [27, 28, 29]); their patterns of query rewriting (e.g. [65, 68]); and the average number of queries to solve an information need (e.g. [27, 28, 29, 30, 31, 33, 64, 68, 78, 79]).

2.2. Query log exploitation

2.2.1 Automatic query suggestion

Much of the earlier work on the exploitation of query logs deals with the computation of inter-query similarities to provide query expansion, suggestion or reformulation. For instance, Beeferman and Berger [6] described a technique to provide suggestions by clustering queries according to the co-occurrence of URLs within the click-through data. A similar work was developed by Wen, Nie and Zang [75], and Wen *et al.* [77]; in this case the clusters were primarily used to assist human editors from a question-answering search engine to detect frequently asked questions.

Cui *et al.* [17, 18] argued that when users click a result they are assessing that document as relevant to the query and, thus, terms highly related to those in the queries can be extracted from the visited documents; thus, they applied that idea to click-through data in order to perform query expansion. Huang, Chien and Oyand [26] performed a similar work to provide users with additional relevant terms for their queries. This later approach introduced some novelties: firstly, the suggested terms did not come from retrieved documents but from other queries similar to those submitted by the user; secondly, their method worked with whole sessions corresponding to unique information needs in contrast to other techniques which usually operate over individual queries.

Jones *et al.* [38] described a rather different method to perform query reformulation within a sponsored search environment. The aim of their technique is to rewrite the original query to better match the relatively small number of advertisers, thus, improving recall without affecting the original intent of the user.

Chien and Immorlica [12] proposed a method to compute inter-query similarity based only on temporal clues; so, it is somewhat complimentary to the aforementioned techniques. The idea underlying their approach is quite simple: two queries are highly related if they tend to co-occur at the same time; therefore, the similarity of two queries is derived from the correlation coefficient of their frequency functions.

2.2.2 Re-ranking of search results

The use of query logs to improve the results of search engines has also become a popular matter of research. Joachims [36] was the first one to employ click-through data as relevance judgments about the results retrieved for each query. He used such data to learn the ranking function of a specialized meta-search engine which later outperformed a commercial search engine. Later, Joachims *et al.* [37] deeply analyzed the issues regarding the use of click-through data as implicit feedback. The use of query logs as a source of implicit information to improve ranking functions has also been studied by Agichtein, Brill and Dumais [2], or Zhand and Dong [83].

2.2.3 Other uses for query logs

Query logs have also been used in other contexts. For instance, Xue *et al.* [82] used them as a source of metadata for the documents appearing within the click-through data. Chuang and Chien [14] proposed a technique to categorize queries submitted to a search engine to assist in the process of building Web taxonomies. Cucerzan and Brill [16] were the first to propose the use of query logs to perform spelling correction. A web page summarization system which exploits the query terms associated with the click-through data was developed by Sun *et al.* [73]. Last, but not least, the analysis of query logs can shed light on low level aspects of search engines such as query caching [43, 80] or index storage [4].

2.3. Privacy issues

The AOL search data scandal exposed above revealed the issues and risks concerning the release of query logs when no measures to preserve users' privacy are taken. In this regard, Xiong and Agichtein [81] proposed two orthogonal dimensions to anonymize query logs while preserving most of their utility for research. The first dimension relates to the granularity of the data, that is, how many log records can be grouped and linked as belonging to the same individual. The second one deals with the de-identification of the queries themselves by removing or generalizing sensitive terms.

Xion and Agichtein proposed five granularity degrees: *user*, *session*, *query session*, *query* and *aggregate*. Coarse-grained levels preserve more useful information than the fine-grained ones but they are also more prone to privacy breaches. As regards with query de-identification they described a spectrum which ranges from full de-identification to no de-identification at all.

When a query log combines no de-identification of queries with user granularity there is a very real risk of attackers detecting actual identities, which was the case with the AOL search data. Then, arguably, the reason why none of the logs released prior to the AOL dataset have raised privacy concerns is the very short time they span (about 24 hours or less). Consequently, it seems that a query log with no de-identification but employing a granularity degree equal or below the session level should provide reasonable levels of privacy for search engine users while still being useful for research purposes. Hence, a reasonable and simple measure could just consist of limiting search logs to one day and de-identifying sensitive information such as e-mail addresses or number sequences.

Nevertheless, by doing this, research on evolving issues (e.g. financial crisis or presidential campaigns) would be unattainable and, in contrast, although it would be much harder for an attacker to disclose actual identities from just 24 hours of data it could still be possible. Hence, more sophisticated methods should be applied in order to further segment the query log data into very short topical sessions (i.e. sequences of queries related to just one single goal or information need) while still covering long periods of time. Such an idea of splitting users' queries according to different "interests" has also been proposed by Adar [1].

2.4. Review of search session definitions

Up to this point, no definition has been stated neither for *session* nor *query session*. In fact, there seems to be no general consensus about them in the literature: they are sometimes used interchangeably and other times with subtle nuances. Likewise, additional terms have been occasionally proposed to refer to the same concepts or to clarify them. Nevertheless, before following with the literature review it is worth noting some concepts underlying the idea of session in search engines.

First of all, searches are not and end *per se* but a way of achieving some goals: to reach a known website, to find a piece of data or to obtain a resource other than information [59]. In some sense, this extends the concept of *information need* coined by Maron and Kuhns [44]. Secondly, searching is a trial-and-error process and, thus, a query is, according to Swanson [74]:

> […] a guess about the attributes a desired document is expected to have […] the response of the system is then used to correct the initial guess for another try.

That way the users gradually refine both their queries and their goals. Spink *et al.* [67] referred to this process as *successive search phenomenon* and defined it as:

> The process of repeatedly searching over time in relation to a specific, but possibly an evolving information problem.

As a consequence, when users interact with a search engine in order to achieve their goals they produce a sequence of queries able of being recorded and subsequently analyzed. Thus, any definition of session should take account of this iterative and evolving nature, in addition to the underlying existence of user goals.

A third significant factor is the way in which search engines collect logs from the queries they receive under the form of HTTP requests. First of all, a way of separating requests issued by one user from those issued by a different one is required. Most of the search engines employ cookies [39] to this extent. Cookies allow search engines to distinguish different users who are sharing one single IP address or to track particular users no matter they connect to the search engine using different IP addresses in subsequent requests. The major drawback of cookies is that users can remove or disallow them. However, most of the queries a search engine receives carry cookie information –according to Silverstein *et al.* [64] over 96% of them– and this explains why they are a rather common and reliable solution to associate each query to a unique single user.

Some search engines use only one cookie to store an alphanumeric user ID while others resort to a second one to also store a session ID. As of mid-2008, Google, AltaVista, AlltheWeb and Baidu seemed to just employ user identifiers while Yahoo, Ask, Live or Exalead utilized several cookies and, thus, stored both user and session identifiers (e.g. Ask used the cookies `wz_uid` and `wz_sid` while Live used `SRCHUID` and `SRCHSESS`). Cookies storing session IDs tend to be temporal; this way, the search engine provides a new session identifier the first time the user issues a query after having previously closed the browser or when s/he has not submitted any query in more than a certain amount of time (from minutes to hours). Hence, depending on the information stored in the cookies the entries in the query log could contain just user IDs or both user and session IDs.

Hence, a session from a search engine point of view can be: (1) the whole sequence of queries issued by one user during one single day; (2) the sequence of queries issued by one user since s/he starts the browser until s/he quits; or (3) a sequence of queries with no more than a few minutes of inactivity between them. The following literature review takes account of these and other views of search session.

As it was stated above, the first in-depth analysis of a search engine query log was conducted by Jansen *et al.* [33]. They did not provide any session definition but they grouped together all the queries from each user appearing in the log file (*"Excite 1997 small"*).

Spink *et al.* [67] coined the term *search episode* which comprises one unique query and the subsequent user actions (e.g. clicking a result, asking for more results or giving relevance judgments). It must be noted that such search episodes would correspond to single records in common query logs including, at most, the query, the timestamp, the page result number, the clicked URL and its position within the results list.

The first clearly stated definition of *session* in search engines is possibly that of Silverstein *et al.* [64]:

> A session is a series of queries by a single user made within a small range of time; a session is meant to capture a single user's attempt to fill a single information need.

Again, it appears the idea of single goals behind the queries issued to search engines. In addition to this, Silverstein *et al.* pointed out that algorithms to detect session boundaries should resolve when a query entails a new information need and, thus, a new session.

Jansen, Spink and Saracevic [32] revisited the *"Excite 1997 small"* query log and provided a definition of session compatible with the implicit one used by Jansen *et al.* [33]:

> A session is the entire series of queries by a user over a number of minutes or hours. A session could be as short as one query or contain many queries.

Such definition was employed by Wolfram [78] and further refined by Jansen and Spink [28]:

> A session is the entire series of queries submitted by a user during one interaction with the web search engine.

The grouping of all the entries from a user into one single session was criticized by He and Göker [24]; their view is quite similar to that of Silverstein *et al.* [64] given that they judged temporal proximity a crucial factor to identify a session:

> […] a group of user activities that are related to each other not only through an evolving information need but also through close proximity in time […] the start and end of a session are the points where the role behind a query changes.

He, Göker and Harper [25] employed that very same definition but, in addition, they introduced the concept of *session shift* referring to any change occurring between two successive search activities from a user. They proposed several patterns suitable to resolve if there exists or not a shift between activities. The patterns applicable to queries are *Generalization*, *Specialization*, *Reformulation*, *Repetition* and *New*. Given that He *et al.* considered that sessions are associated with evolving topics the only pattern implying a session ending is the one labeled *New*; we will later return on their work on session segmentation.

Wen and Zang [76] provided the definition for *query session* as *"made up of the query and the subsequent activities the user performed"*; in their case activity only refers to clicking the documents obtained in response to the query so this definition is equivalent to those by Spink *et al.* [67] and Hansen and Shriver [23].

Jansen and Spink [30] introduced the term *searching episode* and defined it as:

> The period from the first recorded time stamp to the last recorded time stamp on the search engine server from a particular searcher in a particular day.

The approach by Xiong and Agichtein [81] to provide query log anonymization has been already referred. They argued that the grouping of log entries is a critical factor relating to users' privacy and thus they proposed several grouping degrees. For this section's purposes only three are of interest: *session*, *query session* and *query*.

The first considered granularity level, *session*, retains all the information from the original query log except for the user ID which is removed; if there exists a session ID it is preserved. Thus, given the way in which search engines collect the query logs these sessions would comprise queries issued by one user during at most one day although there could exist several sessions because of the inactivity threshold imposed by some search engines.

The *query session* level is equivalent to the definitions in [19, 24, 25, 64]. At this level both user and session IDs are removed and a set of entries are combined into a *mini*-session. To achieve this granularity a segmentation algorithm is always needed.

Xion and Agichtein described yet another level, *query*, which just preserves the query string without any other information. This level is equivalent to those described in [23, 67, 76] and, according to Xion and Agichtein, it does not greatly improve privacy when compared with the previous groupings and, in contrast, throws away most of the utility in the data.

The last session definition is the one provided by Jansen *et al.* [35]:

> [A session is] a series of interactions by the user toward addressing a single information need. [...] one searching episode will be composed of one or more sessions.

Let us remember that a searching episode is the period of time from the first to the last recorded user action in a given day [30] and, therefore, according to Jansen *et al.*, sessions do not necessarily comprise all the queries issued by a user during one "sitting" but, in contrast, such activity will comprise one or more sessions.

After this review it seems clear that the surveyed views of session essentially diverge due to the use of different granularity criteria. Most of the definitions consider that a session corresponds, at most, to the period of time extending from the first to the last recorded query submitted to a search engine by a certain user in a given day. Furthermore, it seems generally accepted that such a period does not always correspond to only one query but to several ones and that many of them are topically related.

Thus, from now onward the terms *searching episode* and *search session* will be used. The first one refers to the actions performed by a particular user within a search engine during, at most, one day. Such a searching episode can comprise one or more sessions where each of these includes one or more successive queries related to one single information need or goal.

Table 1. Different definitions of session in the literature. As it can be seen different authors define sessions at different granularity levels.

# of queries	Length	Denomination	Authors
1	-	Search episode **Session** Query session Query	Spink *et al.* [67] Hansen and Shriver [23] Wen and Zang [76] Xiong and Agichtein [81]
Many	At most 24 hours	**Session**	Jansen *et al.* [33] Jansen, Spink and Saracevic [32] Wolfram [78] Jansen and Spink [28] Xiong and Agichtein [81]
		Searching episode	Jansen and Spink [30]
	Variable (usually small)	**Session**	Silverstein *et al.* [64] He and Göker [24] He, Göker and Harper [25] Jansen *et al.* [35]
		Query session	Xiong and Agichtein [81]

2.5. *Review of session detection methods*

In this section we will survey several methods to segment query logs into search sessions, that is, short sequences of successive queries related to one single goal or information need. As it will be shown two kinds of clues can be exploited alone or combined in order to detect session boundaries: the time gap between queries, or the query reformulation patterns.

2.5.1 Temporal clues for session boundary detection

Users tend to submit bursts of queries for short periods of time and enter afterwards relatively long periods of inactivity. Thus, to detect session boundaries, Silverstein *et al.* [64] suggested applying temporal thresholds. They used a 5 minutes cutoff: if two queries were less than 5 minutes apart they would belong to the same session and otherwise to different sessions. This method is quite popular due to its simplicity and has been widely used with different thresholds: 5 minutes [19, 26], between 10 and 15 minutes [10, 24], and 30 minutes [10, 19, 57].

Murray, Lin and Chowdhury [47] claimed that applying the same threshold to all users is not necessarily appropriate for every user under every circumstance and, hence, they proposed a technique which finds a threshold for each user by means of an algorithm based on Hierarchical Agglomerative Clustering.

2.5.2 Lexical clues for session boundary detection

Other researchers have suggested the idea of using the content of the queries themselves to determine if there exists a topic change and, thus, a session boundary. In this respect, several classifications of search patterns have been proposed such as those of Lau and Horvitz [40]; Spink, Jansen and Özmutlu [65]; or He, Göker and Harper [25]. The later classified search patterns into eight mutually exclusive categories to indicate the relationship between two consecutive queries and aiming to facilitate the detection of sessions. Özmutlu and Çavdur [50] and Jansen *et al.* [35] have employed mostly these same search patterns.

For the purpose of session boundary detection the following patterns are of interest: (1) *Repetition*, (2) *Specialization*, (3) *Generalization*, (4) *Reformulation* and (5) *New*. The first pattern, *Repetition*, means that the second query q_{i+1} is the same as the first query q_i. *Specialization* refers to the fact that the query q_{i+1} deals with the same topic that q_i but seeks more specialized information (e.g. additional terms have been added to the query). *Generalization* refers to the opposite, the query q_{i+1} is on the same topic that q_i but seeks more general information (e.g. some terms have been removed from the original query). In the *Reformulation* search pattern both queries are about the same topic but the user has both added some terms and removed others from the first query and both queries still have some common terms. The last search pattern, *New*, implies that the second query is on a "different" topic that the first one (in fact, that the queries have not any common term).

It must be noticed that search patterns are determined by means of lexical comparisons (i.e. the presence or absence of common terms); therefore, the major problem of this method is the *vocabulary-mismatch problem* [20]; that is, the existence of topically related queries without common terms. For instance, two subsequent queries such as `IR` and `information retrieval` would flag a *New* search pattern when it is likely that both queries are pursuing the same informational goal.

2.5.3 Machine-learning methods to combine temporal and lexical clues

When using search patterns to detect session boundaries there are two major approaches: (1) It can be assumed that the *New* search pattern always implies a session boundary [35] or (2) statistical information can be collected from the query logs to find out the probability that this search pattern actually implies a change of session depending on the time gap between the two successive queries (e.g. He, Göker and Harper [25]). Given the nature of the *New* search pattern the first approach implies that many topically related queries are divided into different sessions; the approach of He *et al.* can partly solve this problem and thus it will be carefully described.

He *et al.* [25] suggested to combine both temporal and search pattern information to decide if two queries belong or not to the same session. For every pair of queries their algorithm computes both the temporal gap and the search pattern. For each time interval and search pattern there exist pre-computed probabilities that are combined by means of the Dempster–Shafer theory. If the resulting value exceeds a certain threshold the algorithm flags a session change and, otherwise, a session continuity.

Apart from the combination of temporal and lexical data the approach by He *et al.* has other two key aspects. Firstly, it requires training data, that is, human judges must analyze a sample from the query log to mark session changes. This information is used afterwards to compute the conditional probability of shift given the time interval and search pattern. Secondly, in order to combine those probabilities into the Dempster's rule two confidence weights are required; both the weights and the aforementioned threshold are to be obtained by means of genetic algorithms. Thus, a measure of performance for the session detection is needed and to this aim He *et al.* chose the F_β measure [58] setting β to 1.5.

Özmutlu and Çavdur [50] replicated that work applying the technique to a sample of about 10,000 queries from Excite and testing different values for the input parameters. They found that (1) the method is dependent on the parameters and (2) the parameters obtained for one particular dataset are not necessarily the most successful ones to segment that dataset. This put into question the general applicability of the technique by He *et al.* although Özmutlu and Çavdur pointed out several reasons for such results: the use of Dempster–

Shafer theory to combine temporal and lexical evidence, the use of genetic algorithms to find the values of the parameters required or even the fitness function chosen for the genetic algorithm.

Hence, Özmutlu and Çavdur stated that the idea of combining both temporal data and search patterns deserved *"more exploration through other methodologies and other evaluation measures"*. Since then, they and their colleagues have revisited the Dempster–Shafer method [53] and studied the feasibility of additional ones: neural networks [51, 52], multiple linear regression [48, 55], Monte–Carlo simulation [49] and conditional probabilities [54].

In addition to He *et al.* and Özmutlu *et al.* a few other researchers have applied machine learning methods to the problem of session detection. For instance, Radlinski and Joachims [57] used SVM classifiers; however these were ultimately dismissed because, according to these researchers, the training was relatively expensive and SVMs hardly improved the results attained using naïve methods (temporal cutoffs).

2.5.4 Heuristic-based methods for session boundary detection

Finally, there are several lesser known segmentation techniques that deserve some attention in order to get a whole picture of the state of the art.

Shen, Tan and Zhai [62] proposed a method which did not employ temporal information but just compared the queries (by means of the cosine similarity). As it was stated before, related queries does not necessarily share common terms and, thus, Shen *et al.* did not compare the actual queries but their expanded representations. Such representation consisted of the titles and snippets for the first 50 results provided by a search engine for every individual query.

Seco and Cardoso [61] described a really simple technique: a candidate query belongs to a new session if it does not have any term in common with the queries from the current session or the time gap between the candidate query and the last query in the current session is larger than 60 minutes.

Shi and Yang [63] developed the so-called *dynamic sliding window segmentation* method which relies on three temporal constraints: α, the maximum time gap between two successive queries belonging to the same session; β, the maximum inactivity time within the same session; and γ, the maximum length for a single session. Shi and Yang empirically set the values for α, β and γ to 5 minutes, 24 hours and 60 minutes, respectively. This means that two successive queries with a gap shorter than 5 minutes should belong to the same session (as long as the whole session is under the 24 hour maximum length) and that two queries with a gap longer than 60 minutes would belong to different sessions. Those queries with a time interval between 5 and 60 minutes are compared using the Levensthein distance to decide if they are similar enough to belong to the same session or not.

```
                                                    | desperately seeking susan audio   2001-02-06 17:46:48
desperately seeking susan audio   2001-02-06 17:46:48 | desperately seeking susan sound   2001-02-06 17:48:33
desperately seeking susan sound   2001-02-06 17:48:33 |
madonna get into the groove       2001-02-06 17:55:47 | madonna get into the groove       2001-02-06 17:55:47
madonna get into the groove       2001-02-06 17:57:29 | madonna get into the groove       2001-02-06 17:57:29
video games cheats and codes      2001-02-06 18:02:56 | madonna get into the groove       2001-02-06 18:11:40
video gameshark codes             2001-02-06 18:10:27 | madonna get into the groove       2001-02-06 18:12:27
madonna get into the groove       2001-02-06 18:11:40 |
madonna get into the groove       2001-02-06 18:12:27 | video games cheats and codes      2001-02-06 18:02:56
                                                    | video gameshark codes             2001-02-06 18:10:27
```

Figure 1. Left: a sequence of queries submitted by one user. Right: sessions obtained using the Buzikasvili's method; notice the way in which intermingled topics are separated (e.g. queries about Madonna and videogames).

Buzikashvili [7, 8], and Buzikashvili and Jansen [10] described an interesting method to not only segment query logs into sessions but to also separate intermingled multitasking queries (see Figure 1). This method operates in two steps: in the first one a time cutoff (15 or 30 minutes) is used to obtain *temporal sessions*; in the second one the queries within those temporal sessions are compared to build a similarity graph for which the transitive closure is computed. All the queries connected in the transitive closure belong to the same session.

3. Research motivation

3.1. Research questions

As it has been exposed, query logs are not only a rich source of information on Web searchers' behavior but they are also useful to improve many different aspects of search engines operation. However, because such files potentially contain a great amount of sensitive personal information there exist serious privacy concerns with regards to the open disclosure of such data.

We have seen that all but one of the freely available query logs contain data for just one day, and that the release of such query log by AOL greatly worried the general public because of its privacy flaws. Arguably, by limiting the available search data to just one day future privacy leaks could be avoided; however, this author maintains that by doing this (1) privacy attacks would be just more difficult but not totally impossible, and (2) research on topics evolving through several days would instead be unfeasible. In fact, session detection methods able to segment query logs into short topical sessions could probably be a much more feasible idea to dispel ethical and privacy concerns while still providing query data spanning several days, even months.

Thus, the main research questions addressed in this study included the following: (1) How should the performance of session detection methods be evaluated? And, (2) which are the most appropriate methods to perform session segmentation on query logs?

It must be said that in addition to the session boundary detection methods studied in the Literature Review this author proposed a new heuristic-based technique. This new method will be described in the following subsection.

3.2. Proposal for a new session detection method

As it has been previously exposed, most of the session detection methods are based on two common assumptions: (1) the larger the time gap between two successive queries, the lesser the likelihood of both queries belonging to the same session and (2) the larger the similarity between two queries, the larger the possibility of both being part of the same session. The methods described in the Literature Review section implement these heuristics in one way or another. The method proposed by the author is based on a "geometric interpretation" of these assumptions. Such interpretation is shown in Figure 2 which, as it can be seen, poses two extreme cases.

The first case, A, is that in which a user submits to the search engine two simultaneous but totally dissimilar queries (e.g. they have no common terms). In absence of more information it makes sense to assume that both queries belong to the same session. Certainly, this scenario could also pertain to a multitasking user [66] or not even to a person but to a robot; however, the author has decided to leave these issues for future research. The B case is just the opposite: queries q_i and q_{i+1} are identical but q_i is issued at the beginning of the session while q_{i+1} just occurs at the very limit of the session time.

Thus, the curve from A to B encloses every conceivable combination of lexical distance and time gap between two queries belonging to the same session. Of course, to implement this "geometric approach" it must be defined (1) what is understood by "lexical distance", (2) the time limit for the sessions and (3) the shape of the curve from A to B which will finally determine the grain of the eventual sessions.

In this work the term "lexical distance" is used instead of the, by far, more common "string metric" due to two reasons. First of all, to explicitly state that the proposed method just relies on the data available in the query logs; that is, it does not employ query expansion techniques. Secondly, to note the method is left deliberately open with regards to the implementation of the inter-query comparison. For instance, queries could be compared by means of common terms or by using classic string metrics; in addition to this, it could be chosen to compare query q_{i+1} only with q_i or with all the previous queries belonging to the current session. The only requirement is that the chosen "lexical distance" must equal zero for identical queries and the unity for totally dissimilar queries. In a later section the actual implementation used by the author in the experiments is described.

In addition to the lexical distance any implementation of the method must establish the so-called *"session time limit"*. The simplest way consists of using an arbitrary threshold T to define the maximum assumed duration of a session. Thus, the time gap t between queries q_i and q_{i+1} can be normalized by just dividing it by T; this way the normalized time gap between two queries potentially belonging to the same session will always be in the interval [0, 1]. For the experiments a threshold of 24 hours has been used.

Therefore, given two queries it is possible to compute a lexical distance normalized in the interval [0, 1] and a time gap also normalized in the same interval. Thus, the case A (simultaneous but dissimilar queries) will correspond to the point (0, 1) while the case B (identical queries occurring at the beginning and the end of the session) will correspond to the point (1, 0). Different curves can be used to define the sessions providing different enclosed areas and, thus, obtaining more coarse or fine-grained sessions.

In short, just like He *et al.* [25] and Özmutlu *et al.* [50], this author considers essential the combination of lexical and temporal data from query logs to perform effective session detection. However, unlike the work developed by those researchers the new approach does not require prior training and can be easily implemented for real-time application. This new method is based on a geometric interpretation which relates two queries both in lexical and temporal dimensions and, thus, resolves if those queries belong or not to the same session. In the following sections the experiments performed, as well as the evaluation methodology and the obtained results will be described.

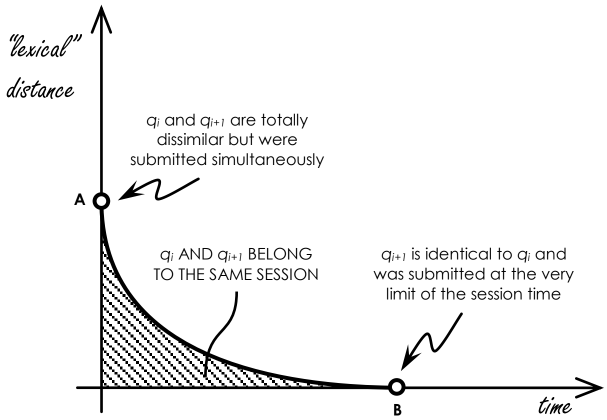
"lexical" distance
q_i and q_{i+1} are totally dissimilar but were submitted simultaneously
A
q_i AND q_{i+1} BELONG TO THE SAME SESSION
q_{i+1} is identical to q_i and was submitted at the very limit of the session time
B
time

Figure 2. Geometric approach to session segmentation. Dissimilar queries submitted simultaneously (case A) and identical queries issued with a time gap equal to the session length (case B) belong to the same topical session. Any other cases enclosed by the curve from A to B also belong to the session (hatched area).

4. Research design

The main goal of this study was to find the most appropriate method to topically sessionize a big query log. To attain that, an evaluation framework was needed, in addition to ground-truth data against which to compare the solutions achieved by the different session detection techniques. In fact, such an evaluation framework (in particular the performance measures to employ) would address the first research question. On the other hand, the results obtained by the different sessionization methods within that evaluation would address the second research question. Consequently, this section describes the datasets employed to prepare the ground-truth data; provides an analysis of such ground-truth files; details the performance measures to evaluate the different methods; and provides implementation details for every evaluated method.

4.1. Data preparation

Seven datasets were used for the evaluation experiments: *"Excite 1997 small"*, *"Excite 1997 large"*, *"Excite 1999"*, *"AlltheWeb 2001"*, *"Excite 2001"*, *"AltaVista 2002"* and *"AOL 2006"*. All of them have been aforementioned in the Literature Review section so just a descriptive summary is provided in Table 2.

For each of those datasets a ground truth file was built; that is, a representative subset of queries was extracted from each log and was manually segmented into topical sessions. Such ground truth files allow researchers to compare the results attained by different methods with those provided by human experts (this approach was followed, for instance, in [21, 25, 50]). To obtain representative, yet still manageable, samples from the query logs systematic sampling was applied. In order to avoid bias towards the most active users the sampling was performed over the user space rather than the query space. The sample size, however, was estimated for the searching episode population in order to obtain samples exhibiting similar features to those of the original data (Figure 3 shows the formula applied to estimate the sample size).

$$ss = \frac{Z^2\sigma^2}{\beta^2}$$

Figure 3. Formula to estimate a representative sample size. Z is the confidence level; it takes values 2.58, 2 and 1.96 for levels of 99%, 95.5% and 95%, respectively. β is the error rate expressed as decimal. σ is the sample standard deviation, if it has not been estimated by means of a pilot sample it can be assumed to be 0.5 –this was the case for the samples obtained from the original datasets.

For instance, the AOL data contains 7,381,505 searching episodes from 657,426 users and, thus, the average number of searching episodes per user is 11.23. If we set the confidence level (*Z*) to 99% and the error rate (β) to 2% we would need a sample of 4,160 searching episodes and, in turn, 378 users. This can seem a fairly small sample but since the AOL data comprises about 30 million queries, these 378 users would mean approximately 17,000 queries which are not very manageable to be manually segmented into sessions. That's why the author finally set the confidence level to a lower value of 95.5% maintaining the error rate to 2%. With these settings the number of searching episodes is reduced to 2,500 which for the AOL data supposed 223 users and about 10,000 individual queries (which are still a daunting segmentation task).

Table 3 shows the actual sample size for each query log; please notice that the aggregate size for all the sample files is about 95,000 queries. To the best of our knowledge, this make it the largest and most exhaustive query log to have been manually prepared to the date given that it is almost ten times larger than those described in [21, 50, 66, 71].

The manual processing of the data was simple but extremely time consuming. A human expert worked through each sample from which temporal data was removed to avoid any bias due to "prejudices" regarding the time elapsed between subsequent queries[2]. This way the expert just had to decide if two queries were or not topically related; in the former case both would belong to the same session and to different sessions in the later. This relatedness, however, was not easy to assess and most of the time the judge had to eventually submit the queries to a search engine to fight the lack of domain-specific knowledge and take a decision.

Table 2. Descriptive information regarding the datasets used in the experiments. The figures commonly reported in the literature appear in brackets. In some cases there are minor discrepancies (e.g. "Excite 1997 small" or "Excite 1997 large") which can be attributed to post-processing artifacts. Other data sets (e.g. "Excite 1999" or "Excite 2001") reveal larger differences. It is also worth noting that some query logs commonly assumed to contain data from one single day span, in fact, more than 24 hours ("Excite 1997 large" and "AltaVista 2002").

	Date of collection	Time span	# of entries	# of different users
Excite 1997 small [33]	9 March 1997	30 minutes	51,474 *(51,473)*	18,107 *(18,113)*
Excite 1997 large [68]	16 September 1997	Approx. 33 hours	1,025,907 *(1,025,910)*	211,047 *(211,063)*
Excite 1999 [27, 79]	20 December 1999	8 hours	2,477,283 *(Over 2.5 million queries)*	537,553 *(Over 200,000)*
AlltheWeb 2001 [70]	6 February 2001	24 hours	1,257,943 *(451,551 queries)*	153,740 *(153,297)*
Excite 2001 [69]	May 2001	24 hours	1,229,282	305,339 *(262,025)*
AltaVista 2002 [31]	8 September 2002	Approx. 27 hours	3,518,498 *(Approx. 3 million records)*	370,585 *(369,350)*
AOL 2006 [56]	From 1 March 2006 to 31 May 2006	92 days	36,389,566 *(36,389,567)*	657,426 *(657,426)*

[2] Temporal data could not be totally removed from the AOL sample because this data spans several days; thus, the timestamps for the first and last query of each day were included.

Table 3. Descriptive information regarding the seven manually segmented samples. Notice that the average number of queries per topical session is very similar across the different samples.

	# of searching episodes	# of users	# of queries	# of topical sessions (manually)	Queries per session
Excite 1997 small	2,155	2,155	6,090	2,619	2.33
Excite 1997 large	2,500	2,500	12,414	4,189	2.96
Excite 1999	2,499	2,499	8,716	3,293	2.65
AlltheWeb 2001	2,500	2,500	20,960	7,471	2.81
Excite 2001	2,500	2,500	9,496	3,414	2.78
AltaVista 2002	2,500	2,500	25,461	9,290	2.74
AOL 2006	2,500	223	11,484	4,254	2.70

4.2. *Data analysis*

Allegedly, all of the query logs contain data from just one day except for *"AOL 2006"* which has records corresponding to 92 days. However, both *"Excite 1997 large"* and *"AltaVista 2002"* contain queries recorded for more than 24 hours (see Table 2). The first one, *"Excite 1997 large"*, contains queries from September 15th 1997 around 23:10 to September 17th 1997 around 7:58; that is, about 33 hours. The second one, *"AltaVista 2002"*, contains queries from September 8th 2002 around 4:00 to September 9th 2002 around 7:00; that is, about 27 hours. This fact could affect those session detection methods employing just lexical clues and, thus, it had to be taken into account before evaluating such techniques.

Additionally, all the query logs are affected to some extent by the inclusion of queries submitted by software agents. In some cases a visual inspection of the data showed that such queries could not have been issued by human users (Figure 4 shows some examples); however, there is not a renowned and publicly available method to tell apart human users from software agents [34].

To the best of author's knowledge the only described methods to filter out queries issued by robots are those by Jansen *et al.* [34, 35], Zhang and Moffat [85] and Buzikashvili [9]. Jansen *et al.* ignored data from users with 100 or more successive queries. Zhang and Moffat removed those users who never click on a search result and Buzikashvili proposed using a temporal window to remove those users issuing too many queries within that period. None of these methods has been thoroughly evaluated and, in addition to this, after an experiment with a 2,500 users sample from the *"AOL 2006"* log this author found that their results have little to no overlap. Thus, the approach by Jansen *et al.* was eventually chosen because it is the most conservative one: it pointed out 0.4% of users as robots in contrast to Buzikashvili's 18.5% and Zhang and Moffat's 11%. Hence, all searching episodes with 100 or more queries were ignored during the evaluation.

```
doubts OR aggregating OR willow OR reeling OR exile                2002-09-08 08:05:29
conquest OR fortified OR provokes OR preempt OR deluge             2002-09-08 08:05:38
tenement OR tiffany OR groves OR pruners OR democracy              2002-09-08 08:06:00
returned OR herd OR signed OR midst OR resorting                   2002-09-08 08:06:24
nominee OR christiansen OR differentiations OR technic             2002-09-08 08:06:48
terms OR interpolating OR mets OR shudder OR unknowns              2002-09-08 08:07:27
alternator OR approbate OR pecuniary OR candler OR polytechnic     2002-09-08 08:07:39
shakable OR postal OR domino OR purify OR fiend                    2002-09-08 08:07:44
occidentalizing OR frug OR revolting OR parasite OR fortify        2002-09-08 08:08:23
willamette OR fortieth OR apostrophe OR query OR germinal          2002-09-08 08:08:57
```

Figure 4. Successive queries from one user in the "AltaVista 2002" query log. The time gap between queries is too short to allow a human an analysis of the results, the queries do not show apparent topical relation, and the use of the `OR` operator can only obtain almost random web pages.

4.3. *Proposed evaluation method*

Once a gold-standard is available the evaluation and comparison of session detection algorithms should be straightforward. However, because this is a rather new evaluation task there is not much literature on the topic. He and Göker [24] were, to the best of our knowledge, the first to suggest the evaluation of session detection methods on the basis of manually prepared data. After that, He, Göker and Harper [25] proposed a set of measures to evaluate such methods against a gold-standard. To attain this, they adapted the well-known measures *precision*, *recall* and *F-measure*. Equations (1) to (3) show their formulations in terms of the number of topic shifts and topic continuations found by the segmentation method (N_{shift} and N_{contin}, respectively); the number of topic shifts found by the human judges (N_{true_shift}); and the number of topic shifts agreed by both the segmentation method and the experts ($N_{shift\&correct}$).

$$P = \frac{N_{shift\&correct}}{N_{shift} + N_{contin}} \tag{1}$$

$$R = \frac{N_{shift\&correct}}{N_{true\_shift}} \tag{2}$$

$$F_{\beta} = \frac{(1+\beta^{2})PR}{\beta^{2}P + R}, \beta = 1.5 \tag{3}$$

Özmutlu and Çavdur [50] provided a corrected version for precision (Equation 4) because of a misprint in He *et al.*'s formulation. This second formulation has been also used by other researchers (e.g. [26, 57]).

$$P = \frac{N_{shift\&correct}}{N_{shift}} \tag{4}$$

In addition to precision, recall and F_{β} measures both He, Göker and Harper [25]; and Özmutlu & Çavdur [50] reported numbers of *Type A* and *Type B* errors. The former occur when queries on the same topic are wrongly divided into two different sessions; the later occur when queries on two different topics are wrongly grouped into a single session. He *et al.* considered *Type B* errors more harmful than *Type A* and, thus, they emphasized recall over precision by setting β to 1.5 in the F-measure formulation shown in Equation (3).

All of these three performance measures are well-known in IR; however, there exist other kind of problems where a hypothesis from a system is compared against a gold-standard, and the experiences with the evaluation of such systems can shed some light to the evaluation of session-segmentation methods. One of such problems is the so-called Chinese word segmentation: that is, the tokenization of Chinese text (which is a run of characters without separating blanks) into a sequence of words.

The similarities between this problem and that of session detection seem clear: the input is a continuous run of items and the system must insert separators in between; then, the system's output is compared against a gold-standard causing four different circumstances: (1) both the system and the judge agree with the inserted blank, (2) the system has not inserted a blank the judge had inserted, (3) the system has inserted a blank the judge had not, and (4) neither the system nor the judge inserted a blank.

The first international bakeoff on Chinese word segmentation was held in 2003 [72]. This event, like all bakeoffs, relies heavily on evaluation and comparison between systems. The performance measures used to the date are recall, precision, balanced F-measure, *recall on out-of-vocabulary words*, and *recall on in-vocabulary words*. The last two measures are not applicable to the problem of session detection, and the first ones were the same used in [25, 50]. Yet, according to Makhoul *et al.* [42] the F score underestimate the importance of deletion (missing blanks) and insertion (spurious blanks) errors. Therefore, they described two performance measures better suited for this kind of segmentation tasks. The first of such measures is the so-called *ERR* as employed by the Message Understanding Conference [13]; Equation (5) shows its formulation adapted for the evaluation of session detection methods (i.e. without considering substitutions).

$$ERR = \frac{D+I}{C+D+I} \tag{5}$$

Makhoul *et al.* argued that this measure still poses one problem which can be seen in Equation (5): the sum of correct (C) and deleted (D) separators equals the number of true topic shifts in the ground truth data. Because this is constant for a given evaluation process, the *ERR* measure depends linearly on the number of deletions (D) which appears in the numerator of the formulation and non-linearly on the number of insertions (I) which appears both in the numerator and the denominator. This way, deletion errors increase *ERR* by a bigger amount than insertion errors. To avoid this problem Makhoul *et al.* proposed a new measure named *SER* which is shown in Equation (6) adapted to our particular context.

$$SER = \frac{D+I}{C+D} \tag{6}$$

These measures, in special *SER*, provide a more accurate (and fairer) sense of system performance for this kind of problems (i.e. segmentation tasks). However, there are some drawbacks: (1) the *ERR* measure underestimates insertion errors, (2) the *SER* measure can be greater than 1 for highly error-prone systems, which Makhoul *et al.* considered "unaesthetic", and (3) use of both *ERR* and *SER* must fight against the inertia of using precision, recall and the *F*-measure to evaluate performance.

The following equations show the final formulations for precision, recall, *F*-measure, *ERR* and *SER* finally applied to perform the evaluation described in the results section. Please notice that precision and recall are those by [25, 50] and that the *F*-measure is balanced.

$$P = \frac{C}{C+I} = \frac{N_{shift\&correct}}{N_{shift}} \tag{7}$$

$$R = \frac{C}{C+D} = \frac{N_{shift\&correct}}{N_{true\_shift}} \tag{8}$$

$$F = \frac{C}{C+\frac{1}{2}D+\frac{1}{2}I} = \frac{N_{shift\&correct}}{N_{shift\&correct}+0.5(N_{true\_shift}-N_{shift\&correct})+0.5(N_{shift}-N_{shift\&correct})} = 2\frac{N_{shift\&correct}}{N_{true\_shift}+N_{shift}} \tag{9}$$

$$ERR = \frac{D+I}{C+D+I} = \frac{N_{true\_shift}-N_{shift\&correct}+N_{shift}-N_{shift\&correct}}{N_{true\_shift}+N_{shift}-N_{shift\&correct}} = \frac{N_{true\_shift}+N_{shift}-2N_{shift\&correct}}{N_{true\_shift}+N_{shift}-N_{shift\&correct}} \tag{10}$$

$$SER = \frac{D+I}{N_{true\_shift}} = \frac{N_{true\_shift}+N_{shift}-2N_{shift\&correct}}{N_{true\_shift}} \tag{11}$$

4.4. Session detection using multiple methods

As it has been shown, there exists a rich literature about session detection in query logs; however, most of the methods have not been thoroughly evaluated or the performance results are not comparable because they were obtained on test collections which are neither available nor replicable. Thus, this author decided to re-implement all the aforementioned methods and his own technique and run them on the same gold-standard to obtain comparable performance measures. Because all of the techniques have been introduced in previous sections this one will just provide a few implementation details.

Table 4. Information regarding the different sessionization methods evaluated in this study. Some of them have alternative names by their respective authors. The kind of information employed to perform the segmentation is shown.

Method	Time data	Lexical data
Temporal [24, 26, 64] (described as *Method 2* in [35])	✓	
AgglomerativeClustering (described as *HAC* in [47])	✓	
QueryContent (described as *Method 3* in [35])		✓
QueryContentExpanded (a variant of *QueryContent* devised by the author)		✓
UCAIR [62]		✓
DempsterShafer [25, 50, 53]	✓	✓
LinearRegression [48, 55]	✓	✓
ConditionalProbs [54]	✓	✓
MonteCarlo [49]	✓	✓
SecoCardoso [61]	✓	✓
DynSlidingWindow (described as *Dynamic Sliding Window Segmentation* in [63])	✓	✓
Buzikashvili [7, 8, 10]	✓	✓
Geometric (proposed by this author)	✓	✓

4.4.1 Methods relying on temporal clues

Two methods employ only temporal information: Temporal and AgglomerativeClustering. The Temporal technique applies a topic shift between two successive queries from the same user when the time gap between both is longer than 30 minutes; otherwise there is a topic continuation. This threshold was employed in [10, 19, 35, 57]; no other thresholds were used for the experiments described in this paper. The AgglomerativeClustering is a direct implementation of the technique proposed by Murray *et al.* [47].

4.4.2 Methods relying on lexical clues

Three methods do not employ temporal data but only lexical information: QueryContent, QueryContentExpanded and UCAIR. The first of them was described as *Method 3* by Jansen *et al.* [35]; this algorithm applies a topic shift when two successive queries from the same user have no common terms.

The QueryContentExpanded algorithm is identical to the QueryContent method but queries are replaced by expanded representations obtained in a way similar to that of [15, 45, 60, 62]: First, the query is issued to a search engine, then the titles and snippets from the first page of results are concatenated and finally the most frequent terms are extracted (removing stop-words). Additionally, if the top result is a Wikipedia article then its first 4KB of data are appended to the snippets before extracting the keywords. In addition to these keywords all the terms from the original query are also included.

The third and last lexical sessionization method is the so-called UCAIR described by Shen *et al.* [62]. Their method is similar to QueryContentExpanded because they also rely on a search engine to expand the queries; however, the way in which query similarity is assessed by QueryContentExpanded is rather naïve when compared to the UCAIR approach. This technique operates in the following phases: (1) Titles and snippets from the first 50 results for each query are obtained; (2) a vector for each result is computed using pivoted *tf·idf*; and (3) these vectors are aggregated into a unique centroid vector for each query. Hence, to compare two queries the cosine similarity for both centroid vectors is computed; if it exceeds a predefined threshold the queries are considered to belong to the same session and to different sessions otherwise. It must be noted, however, that Shen *et al.* did not provide the threshold they employed for their experiments and, thus, some educated guesses had to be done in order to implement their technique.

4.4.3 Methods relying on both temporal and lexical clues

The rest of the evaluated methods employ both temporal and lexical information to perform the segmentation. Some of these methods require prior training and can be considered machine learning methods (e.g. DempsterShafer or LinearRegression). To evaluate all of these methods the parameters reported by the original authors were used.

The method DempsterShafer replicates the technique originally proposed by He, Göker and Harper [25]. These authors suggested to apply the Dempster–Shafer theory to combine temporal and lexical information to decide if two successive queries belong or not to the same topical session. The temporal information consists of the time gap between the queries while the lexical information is the corresponding search pattern exhibited by the queries (i.e. *New*, *Reformulation*, *Specialization*, or *Generalization*). To apply this technique several probabilities and parameters must be provided and, thus, the settings described in [50, 53] were used for the experiments.

Özmutlu *et al.* explored different ideas to combine temporal and lexical information and proposed three methods, namely, ConditionalProbs, MonteCarlo and LinearRegression. The first one, ConditionalProbs [54] is quite simple: the method computes the time gap between two successive queries and their corresponding search pattern; then, it searches in a table for the probabilities of topic shift and topic continuation conditioned on that particular time gap and search pattern; finally, the method always chose the option with the largest probability. It must be noticed that a *post-hoc* analysis of this method by their authors reduced it to a heuristic method: *there exists a topic shift if the time gap between queries is equal or greater than 30 minutes and both queries do not share any common term*. Interestingly, the SecoCardoso method [61] is comparable to this heuristic interpretation although not totally equivalent: *there exists a topic shift if the time gap between two queries is larger than 60 minutes or the candidate query and the current session do not have any term in common*.

Özmutlu and Buyuk [49] further elaborated the idea of using conditional probabilities by means of Monte–Carlo simulation. To decide if there exist a topic shift or a topic continuation between two queries they still rely on the conditional probabilities but, instead of choosing the largest probability, they produce 10 random numbers in the [0, 1] range which can be considered as "votes" for or against inserting a topic shift.

Özmutlu [48]; and Özmutlu, Özmutlu and Spink [55] applied multiple linear regression to find if the common assumption about the dependency of topic shift on time gaps and search patterns has got or not any real basis. They concluded that time gap, search pattern and query position within the session have an actual effect on topic shifts and, as a side effect, they proposed to use the regression equation as the basis for a segmentation technique which is reproduced in the LinearRegression method.

With regards to the lesser-known methods, the DynSlidingWindow technique by Shi and Yang [63] is readily reproducible given that they provide a very detailed algorithm. In contrast, the method proposed by Buzikashvili [7, 8] is not totally obvious and some guesswork was needed in addition to an important adaptation to allow the method to be properly evaluated. As it was aforementioned, the technique proposed by Buzikashvili not only detects session boundaries but untangles mixed multitasking searches; this means that the output of his algorithm reorders the queries (see prior Figure 1) and, thus, it is not possible to evaluate fairly such output against the ground truth files. Hence, it was decided that the finally implemented Buzikashvili method would not reorder the queries.

This segmentation method works in two stages. In the first one the data is segmented by using a time cutoff (15 and 30 minutes were proposed by the original author). In the second one the queries belonging to each temporal session are compared to each other to build a similarity graph. The transitive closure of this similarity graph is computed and the queries connected within it are assumed to belong to the same session.

Buzikashvili provided little details about the actual way in which query comparison is performed: queries are lower-cased and blanks, auxiliary words and endings are removed; it also seems that the similarity measure relies on character *n*-grams but the size and threshold applied are unknown. Thus, the version implemented by this author operates as follows: queries are lower-cased, stop words are removed, and the remaining terms are stemmed. Then, blanks are removed and character *n*-grams are obtained. Finally, two queries are considered similar if they have at least one *n*-gram in common. In addition to this, the output is not reordered; that is, two queries belong to the same session if they are connected in the transitive closure and they are not separated by queries from a different session.

Finally, the so-called Geometric method is the technique proposed by this author. As it was explained, this method also relies on temporal and lexical information to find a topic shift or a topic continuation. In both cases the information is a normalized distance in the [0, 1] range. Thus, the "temporal distance" between queries q_i and q_{i+1} is computed according to equation (12) where t_i and t_{i+1} are the corresponding time stamps for both queries and *time_limit* is a user defined threshold (for this experiments 24 hours).

$$T_{distance} = \frac{t_{i+1} - t_i}{time\_limit} \tag{12}$$

To compute the "lexical distance" the method represents queries and sessions as "bags of character *n*-grams" rather than "bags of words". This approach poses two main advantages [11]: (1) it is noisy tolerant (i.e. performs well in presence of typos, absence of separators between terms, use of separators other than blanks, etc.) and (2) it behaves as a kind of simple stemming method. Other researchers have also applied this technique to compare queries (e.g. [7, 8, 84, 85]).

Hence, as the algorithm groups queries together it builds a vector of *n*-grams for the session. Each time a query is evaluated, its *n*-gram representation is compared with the *n*-grams from the session and a ratio is computed. This ratio will be 0 if there are not any *n*-grams in common between the session and the candidate query and 1 if all the *n*-grams from the query are already present in the session. Then, the ratio is transformed into a distance by subtracting it from the unit.

Thus, for every pair of successive queries the `Geometric` method obtains two values in the [0, 1] interval which, thus, define a point in 2D space. As it was said, this method requires an area to be defined so that all the points which lie within indicate topic continuations. For these experiments the area enclosed by both positive semi axes and a unit circle centered at the point (1, 1) was employed (see Figure 5).

This method requires a prior trivial segmentation step. As it was said, there are three datasets containing information spanning several days, being the most relevant *"AOL 2006"*. Many queries in these logs were issued during daytime but there also exist users which start searching late at night and go on searching after midnight. Hence, such queries appear with two different date stamps although they actually belong to the same "day". To avoid these false shifts the queries are split into different days using a 30 minutes threshold before applying the `Geometric` method. This way, first queries in the early morning can be associated with the last queries issued just before the midnight of the previous day. In contrast, queries issued much later in the day are not associated and, thus, they appear with the correct date stamp.

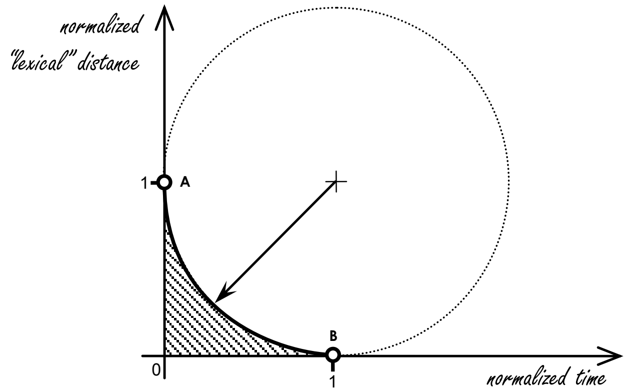

Figure 5. Curve employed for the described implementation of the `Geometric` method.

5. Results

This study was driven by two research questions. The first one deals with a way to properly evaluate session detection methods and it has been addressed in the Research Design section where several performance measures have been discussed. The second research question aims to find the most appropriate methods to detect topical sessions in query logs and will be addressed in this section.

The results obtained with each of the session detection methods are shown below in several tables. For each one *micro-* and *macro-averaged* results are provided. According to Lewis [41] these two ways of aggregating evaluation results emphasize different aspects of the methods to evaluate. When macro-averaging, each individual experiment is considered separately and the average precision (or recall) is computed from the individual precision (or recall) figures obtained within each experiment. When micro-averaging results the data from different experiments is considered to belong to one unique larger experiment. Thus, with regards to the evaluation described in this paper, macro-averaging consists in computing the average measures from those obtained in every individual experiment; micro-averaging consists in counting the total amount of true shifts and errors within all the experiments to compute a single performance figure.

Consequently, micro-averaged results emphasize global performance which in these experiments is dominated by the AOL data which amounts to one third of the total queries in the samples. In contrast, macro-averaged results emphasize the consistent performance of each technique on different datasets. To provide an accurate picture for each method the aggregated results are computed both including and not including the AOL sample. In order to compute the macro-averaged values for *ERR* and *SER* the following equations where applied:

$$ERR = \frac{P + R - 2PR}{P + R - PR} \tag{13}$$

$$SER = \frac{1 + R}{P - 2R} \tag{14}$$

Hence, thirteen different sessionization methods were evaluated. Two of the methods rely on temporal data (`Temporal` and `AgglomerativeClustering`), three on lexical clues (`QueryContent`, `QueryContentExpanded` and `UCAIR`), other four rely on both sources of information and require previous training (`DempsterShafer`, `LinearRegression`, `MonteCarlo`, and `ConditionalProbs`), and the remaining methods employ both temporal and lexical information but can be considered heuristic based (`SecoCardoso`, `DynSlidingWindow`, `Buzikashvili` and `Geometric`).

The so-called `Temporal` method (see Table 5) was taken as a baseline for this session detection task given that it is not only the simplest but also one of the most commonly employed in the literature. It must be noticed that the *"Excite 1997 small"* dataset was not used when evaluating `Temporal` because that log contains data for just half an hour and, thus, using a 30 minute cutoff no topic shifts would be found. With regards to the second method based on temporal data, `AgglomerativeClustering` (see Table 6), the results obtained were quite poor when compared with the baseline.

After evaluating the two methods employing only temporal data it came the turn to those relying just on lexical information (i.e. `QueryContent`, `QueryContentExpanded` and `UCAIR`). As it was previously explained all of them assume that the data belongs to the same day and, thus, they do not perform any prior segmentation of the queries. When comparing the performance of such methods with and without that temporal segmentation it was found, not surprisingly, that the only data showing a noticeable effect was that from *"AOL 2006"*. However, the improvements in performance within that dataset were spectacular when segmenting it into several days and hence all the performance results shown in this paper for such algorithms assume a segmentation of the query logs into several days prior to the application of the corresponding session detection method. As it can be seen from Tables 7, 8 and 9 all of these three methods outperform the baseline in every aggregated performance measure. With regards to the `UCAIR` method, it must be noticed that the author experimented with several thresholds to compare queries because Shen *et al.* [62] did not detail the one they used. The results provided in this paper correspond to a 0.05 threshold which seemed to be the best for the evaluated logs.

The remaining techniques rely both on temporal and lexical information to detect topic shifts. Several of them require a prior "training" phase and, thus, have been evaluated with different parameterizations described in the literature.

Hence, `DempsterShafer` was tested under five different configurations detailed in [50, 53]. According to Özmutlu and Çavdur [50] the parameters obtained for one particular dataset are not necessarily the most successful ones to segment that dataset and the results obtained by this author confirm this claim. Thus, although this technique requires training it does not achieve the best performance on the trained log and, hence, most of its justification is weak. Anyway, the results achieved with the best performance configuration are provided in Table 10 and they show that this method outperforms the baseline in some cases while underperforms in others.

Another method which requires parameters to be obtained from manually segmented data and still seems to underperform the baseline is `LinearRegression`. Apparently, it introduces extremely few topic shifts when compared with the other techniques (see Table 11) which contrast with the reports by the original authors showing an overestimation of topic shifts. Arguably, training on this author's data could shed some light on this issue but this will be left for future research.

The `MonteCarlo` method was tested with different probability sets provided by Özmutlu and Buyuk [49] in addition to probabilities computed by this author from his manually segmented files. These last ones were applied individually to each particular log (i.e. to segment the *"AOL 2006"* sample the probabilities from these data were used). Such configuration obtained, not surprisingly, the best results and thus it can be considered as a top-line showing the best performance that this method is able to achieve (see Table 12). It must be said, however, that the results are quite mixed because this method slightly outperforms the baseline in some cases but underperforms in others.

With regards to the `ConditionalProbs` technique, as it was explained before, it is totally equivalent to a heuristic based approach where a topic shift is flagged whenever the time interval between two successive is above a certain threshold, 30 minutes, and the queries do not contain common terms [54]. Unsurprisingly this method slightly outperforms the baseline (see Table 13). The other heuristic-based methods, namely `SecoCardoso` and `DynSlidingWindow`, both outperform the baseline (see Tables 14 and 15).

One of the most difficult to reproduce techniques was that by Buzikashvili [7, 8, 10] because little implementation details were provided by its original author. Thus, the final implementation of the `Buzikashvili` method requires two parameters: a temporal threshold and the size of the *n*-grams to use when comparing queries. Several parameterizations were evaluated and it was found that the most appropriate for the samples used in these experiments were 30 minutes and trigrams. In addition to this, the method requires queries to be lower-cased, terms stemmed and stop-words removed. This method clearly outperforms the baseline (see Table 16).

The `Geometric` technique proposed by this author was the last to be evaluated and, as it can be seen in Table 17, this method also outperforms the baseline.

Table 5. Performance of the `Temporal` method. Figures for aggregated results are provided in addition to the best and worst results obtained by the method. Results labeled as 1997-2002 do not include the AOL sample while those labeled as 1997-2006 do. N_{true_shift} is the number of topic shifts in the ground truth files; N_{shift} is the number of topic shifts flagged by the method while $N_{shift\&correct}$ is the number of flagged shifts that are correct. *Type A* errors are insertion errors, that is, incorrectly flagged topic shifts while *Type B* errors are deletion errors, that is, correct topic shifts which were not flagged. Two different β values are set for the *F*-measure: by setting β to 1, precision and recall (or which is the same, type A and B errors) are considered equally important; by setting it to 1.5 type B errors are emphasized [24].

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta=1$	F_β $\beta=1.5$	ERR	SER
Best result (AOL 2006)	4039	3382	2985	397	1054	0.8826	0.7390	0.8045	0.7780	0.3271	0.3592
Worst result (Excite 1997 large)	1126	594	334	260	792	0.5623	0.2966	0.3884	0.3471	0.7590	0.9343
Micro-averaged 1997-2002	7328	3311	2191	1120	5137	0.6617	0.2990	0.4119	0.3597	0.7406	0.8538
Micro-averaged 1997-2006	11367	6693	5176	1517	6191	0.7733	0.4554	0.5732	0.5213	0.5983	0.6781
Macro-averaged 1997-2002						0.6330	0.2931	0.4006	0.3511	0.7495	0.8769
Macro-averaged 1997-2006						0.6746	0.3674	0.4757	0.4273	0.6879	0.8098

Table 6. Performance of the `AgglomerativeClustering` method.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta=1$	F_β $\beta=1.5$	ERR	SER
Best result (AOL 2006)	4039	7930	3851	4079	188	0.4856	0.9535	0.6435	0.7355	0.5256	1.0564
Worst result (Excite 1997 small)	338	1722	223	1499	115	0.1295	0.6598	0.2165	0.2919	0.8786	4.7751
Micro-averaged 1997-2002	7097	18994	4044	14950	3053	0.2129	0.5698	0.3100	0.3759	0.8166	2.5367
Micro-averaged 1997-2006	11136	26924	7895	19029	3241	0.2932	0.7090	0.4149	0.4936	0.7383	1.9998
Macro-averaged 1997-2002						0.2016	0.5973	0.3015	0.3725	0.8225	2.7676
Macro-averaged 1997-2006						0.2422	0.6482	0.3527	0.4276	0.7859	2.3798

Table 7. Performance of the `QueryContent` method. Data was previously segmented into different days

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta=1$	F_β $\beta=1.5$	ERR	SER
Best result (AOL 2006)	4039	4713	3807	906	232	0.8078	0.9426	0.8700	0.8965	0.2301	0.2818
Worst result (Excite 1997 small)	338	608	273	335	65	0.4490	0.8077	0.5772	0.6483	0.5944	1.1834
Micro-averaged 1997-2002	7097	10208	6146	4062	951	0.6021	0.8660	0.7103	0.7631	0.4492	0.7064
Micro-averaged 1997-2006	11136	14921	9953	4968	1183	0.6670	0.8938	0.7639	0.8091	0.3820	0.5524
Macro-averaged 1997-2002						0.5651	0.8491	0.6786	0.7354	0.4865	0.8043
Macro-averaged 1997-2006						0.5998	0.8624	0.7075	0.7600	0.4526	0.7131

Table 8. Performance of the `QueryContentExpanded` method. Data was previously segmented into different days.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta=1$	F_β $\beta=1.5$	ERR	SER
Best result (AOL 2006)	4039	4547	3752	795	287	0.8252	0.9289	0.8740	0.8943	0.2238	0.2679
Worst result (Excite 1997 small)	338	544	270	274	68	0.4963	0.7988	0.6122	0.6727	0.5588	1.0118
Micro-averaged 1997-2002	7097	9350	5997	3353	1100	0.6414	0.8450	0.7293	0.7698	0.4261	0.6274
Micro-averaged 1997-2006	11136	13293	9186	4107	1950	0.6910	0.8249	0.7521	0.7785	0.3974	0.5439
Macro-averaged 1997-2002						0.6050	0.8297	0.6997	0.7446	0.4618	0.7120
Macro-averaged 1997-2006						0.6365	0.8438	0.7256	0.7669	0.4306	0.6382

Table 9. Performance of the `UCAIR` method with a 0.05 threshold. Data was previously segmented into different days.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	4690	3787	903	252	0.8075	0.9376	0.8677	0.8933	0.2337	0.2860
Worst result (Excite 1997 small)	338	538	264	274	74	0.4907	0.7811	0.6027	0.6608	0.5686	1.0296
Micro-averaged 1997-2002	7097	9625	6080	3545	1017	0.6317	0.8567	0.7272	0.7721	0.4287	0.6428
Micro-averaged 1997-2006	11136	14315	9867	4448	1269	0.6893	0.8860	0.7754	0.8145	0.3669	0.5134
Macro-averaged 1997-2002						0.5964	0.8398	0.6975	0.7461	0.4645	0.7285
Macro-averaged 1997-2006						0.6265	0.8538	0.7227	0.7681	0.4342	0.6551

Table 10. Performance of `DempsterShafer` using the configuration parameters by Özmutlu, Çavdur and Özmutlu [53].

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	5163	3799	1364	240	0.7358	0.9406	0.8257	0.8664	0.2969	0.3971
Worst result (Excite 1997 small)	338	746	283	463	55	0.3794	0.8373	0.5221	0.6105	0.6467	1.5325
Micro-averaged 1997-2002	7097	11902	6353	5549	744	0.5338	0.8952	0.6688	0.7408	0.4976	0.8867
Micro-averaged 1997-2006	11136	17065	10152	6913	984	0.5949	0.9116	0.7200	0.7833	0.4375	0.7091
Macro-averaged 1997-2002						0.4966	0.8824	0.6355	0.7122	0.5342	1.0120
Macro-averaged 1997-2006						0.5308	0.8907	0.6652	0.7369	0.5017	0.8967

Table 11. Performance of `LinearRegression` using the configuration parameters by Özmutlu [48].

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (Excite 1997 small)	338	8	4	4	334	0.5000	0.0118	0.0231	0.0169	0.9883	1.0000
Worst result (Excite 1999)	794	66	8	58	786	0.1212	0.0101	0.0186	0.0140	0.9906	1.0630
Micro-averaged 1997-2002	7097	515	96	419	7001	0.1864	0.0135	0.0252	0.0189	0.9872	1.0455
Micro-averaged 1997-2006	11136	749	188	561	10948	0.2510	0.0169	0.0316	0.0237	0.9839	1.0335
Macro-averaged 1997-2002						0.2321	0.0150	0.0282	0.0211	0.9857	1.0347
Macro-averaged 1997-2006						0.2551	0.0161	0.0303	0.0227	0.9846	1.0310

Table 12. Performance of the top-line version of `MonteCarlo`.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	3443	3066	377	973	0.8905	0.7591	0.8196	0.7952	0.3057	0.3342
Worst result (AltaVista 2002)	1486	3181	1345	1836	141	0.4228	0.9051	0.5764	0.6700	0.5951	1.3304
Micro-averaged 1997-2002	7097	10902	5621	5281	1476	0.5156	0.7920	0.6246	0.6799	0.5459	0.9521
Micro-averaged 1997-2006	11136	15142	9033	6109	2103	0.5966	0.8112	0.6875	0.7303	0.4762	0.7374
Macro-averaged 1997-2002						0.5641	0.5672	0.5656	0.5662	0.6057	0.8712
Macro-averaged 1997-2006						0.6107	0.5946	0.6025	0.5994	0.5688	0.7845

Table 13. Performance of `ConditionalProbs`. The "Excite 1997 small" dataset was not included because of the 30 minute threshold.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	2874	2737	137	1302	0.9523	0.6776	0.7918	0.7436	0.3446	0.3563
Worst result (Excite 1999)	794	275	210	65	584	0.7636	0.2645	0.3929	0.3311	0.7555	0.8174
Micro-averaged 1997-2002	6759	2540	2106	434	4653	0.8291	0.3116	0.4530	0.3857	0.7072	0.7526
Micro-averaged 1997-2006	10798	5414	4843	571	5955	0.8945	0.4485	0.5975	0.5298	0.5740	0.6044
Macro-averaged 1997-2002						0.8064	0.2999	0.4372	0.3717	0.7202	0.7721
Macro-averaged 1997-2006						0.8307	0.3629	0.5051	0.4389	0.6621	0.7111

Table 14. Performance of `SecoCardoso`.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	5370	3970	1400	69	0.7393	0.9829	0.8439	0.8924	0.2701	0.3637
Worst result (Excite 1997 small)	338	673	279	394	59	0.4146	0.8254	0.5519	0.6325	0.6189	1.3402
Micro-averaged 1997-2002	7097	11069	6299	4770	798	0.5691	0.8876	0.6935	0.7572	0.4692	0.7846
Micro-averaged 1997-2006	11136	16439	10269	6170	867	0.6247	0.9221	0.7448	0.8043	0.4066	0.6319
Macro-averaged 1997-2002						0.5311	0.8729	0.6604	0.7286	0.5071	0.8979
Macro-averaged 1997-2006						0.5608	0.8886	0.6876	0.7531	0.4760	0.8073

Table 15. Performance of `DynSlidingWindow`.

	N_{true_shift}	N_{shift}	$N_{shift\&correct}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	3852	3399	453	640	0.8824	0.8415	0.8615	0.8537	0.2433	0.2706
Worst result (Excite 1997 small)	338	161	96	65	242	0.5963	0.2840	0.3848	0.3386	0.7618	0.9083
Micro-averaged 1997-2002	7097	5624	3864	1760	3233	0.6871	0.5445	0.6075	0.5816	0.5637	0.7035
Micro-averaged 1997-2006	11136	9476	7263	2213	3873	0.7665	0.6522	0.7047	0.6836	0.4559	0.5465
Macro-averaged 1997-2002						0.6548	0.4990	0.5664	0.5384	0.6050	0.7641
Macro-averaged 1997-2006						0.6873	0.5479	0.6097	0.5844	0.5614	0.7014

Table 16. Performance of the `Buzikashvili` method using a 30 minutes threshold and 3-grams.

	$N_{\text{true\_shift}}$	N_{shift}	$N_{\text{shift\&correct}}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	4563	3778	785	261	0.8280	0.9354	0.8784	0.8995	0.2168	0.2590
Worst result (Excite 1997 small)	338	478	252	226	86	0.5272	0.7456	0.6176	0.6613	0.5532	0.9231
Micro-averaged 1997-2002	7097	8997	5723	3274	1374	0.6361	0.8064	0.7112	0.7450	0.4482	0.6549
Micro-averaged 1997-2006	11136	13560	9501	4059	1635	0.7007	0.8532	0.7694	0.7996	0.3747	0.5113
Macro-averaged 1997-2002						0.6044	0.7889	0.6844	0.7211	0.4798	0.7276
Macro-averaged 1997-2006						0.6363	0.8098	0.7126	0.7471	0.4464	0.6531

Table 17. Performance of the `Geometric` method.

	$N_{\text{true\_shift}}$	N_{shift}	$N_{\text{shift\&correct}}$	Type A errors	Type B errors	P	R	F_β $\beta = 1$	F_β $\beta = 1.5$	ERR	SER
Best result (AOL 2006)	4039	4392	3809	583	230	0.8673	0.9431	0.9036	0.9184	0.1759	0.2013
Worst result (Excite 1997 small)	338	495	253	242	85	0.5111	0.7485	0.6074	0.6549	0.5638	0.9675
Micro-averaged 1997-2002	7097	8647	5837	2810	1260	0.6750	0.8225	0.7415	0.7707	0.4108	0.5735
Micro-averaged 1997-2006	11136	13039	9646	3393	1490	0.7398	0.8662	0.7980	0.8229	0.3361	0.4385
Macro-averaged 1997-2002						0.6380	0.8039	0.7114	0.7443	0.4479	0.6522
Macro-averaged 1997-2006						0.6708	0.8237	0.7394	0.7697	0.4134	0.5806

6. Discussion

I now return to the second research question regarding the most appropriate session detection method. As it has been shown, most of them outperform the baseline (a 30 minute cutoff) for every performance measure. The only exceptions are those requiring prior training. It is possible that such poor results can be attributed to the lack of training on the data to segment and, thus, this issue will be left for future research.

Hence, the methods to be compared and their pros and cons analyzed are the following: `QueryContent`, `QueryContentExpanded`, `UCAIR`, `ConditionalProbs`, `SecoCardoso`, `DynSlidingWindow`, `Buzikashvili`, and `Geometric`. All the comparisons will rely on aggregated performance measures (i.e. *F* score, *ERR* and *SER*).

In light of the results shown in Tables 18 to 21 it seems clear that the only method which consistently outperforms the others for virtually every performance measure is `Geometric`. However, `QueryContentExpanded` and `Buzikashvili` are comparable except for their lower performance with regards to *ERR* and *SER* error rates. `ConditionalProbs`, `SecoCardoso` and `DynSlidingWindow` are far from the best results but, in any case, they outperform the baseline. `UCAIR` and `QueryContent` appear in middle positions.

It must be noticed, however, that the best or worst performance of each method cannot simply be attributed to the clues it uses to perform session detection although none method solely relying in temporal data outperforms the baseline, which it was to be expected. Thus, other constrains affecting each method must be considered.

For instance, both `QueryContentExpanded` and `UCAIR` require submitting queries to a search engine which results in extremely long run times and, in addition to this, researchers would be limited by the terms of use of the different available search APIs[3].

Other aspect to consider is the possibility to work on a stream of queries rather than on a query log. The first option is preferable since it dismisses many privacy concerns because the number of queries grouped for a given individual at any moment is much shorter than when collecting a log file. Except for `Buzikashvili`, the rest of the discussed methods can be run in both modes of operation.

With all of this in consideration it seems that the most sensible method to segment query streams in real time is `Geometric` with many of the other methods well behind it in terms of performance (e.g. `QueryContent`, `DynSlidingWindow`, `SecoCardoso` and `ConditionalProbs`). With regards to be run on query logs the most appropriate method is again `Geometric` with `Buzikashvili` as a second best option but with a slightly poorer performance.

[3] Most search APIs limit the number of queries a user can submit per day or the number of available results. For instance, the deprecated SOAP service by Google just allows 1,000 queries per day while the AJAX service does not impose such a limit but provides just a few results. Microsoft Live allows 25,000 queries per day and Yahoo! 5,000. It seems that the new BOSS service by Yahoo! will not have a fixed limit but this would not probably be applicable to automated queries such as those issued by the methods described in this paper.

Table 18. Comparison of each method's performance with the top achiever for every measure (micro-averaged results excluding the *"AOL 2006"* sample).

	$F_{\beta}.\ \beta = 1$	$F_{\beta}.\ \beta = 1.5$	ERR	SER	$\Delta F_{\beta}.\ \beta = 1$	$\Delta F_{\beta}.\ \beta = 1.5$	ΔERR	ΔSER
Buzikashvili	**0.7112**	**0.7450**	0.4482	0.6549	**-4.1%**	**-3.5%**	9.1%	14.2%
CondProbs (5 mins)	0.6069	0.5758	0.5643	0.6893	-18.2%	-25.4%	37.4%	20.2%
DynSlidingWindow	0.6075	0.5816	0.5637	0.7035	-18.1%	-24.7%	37.2%	22.7%
Geometric	**0.7415**	**0.7707**	**0.4108**	**0.5735**	**0.0%**	**-0.2%**	**0.0%**	**0.0%**
QueryContent	**0.7103**	**0.7631**	0.4492	0.7064	**-4.2%**	**-1.2%**	9.3%	23.2%
QueryContentExpanded	**0.7293**	**0.7698**	**0.4261**	0.6274	**-1.6%**	**-0.3%**	**3.7%**	9.4%
SecoCardoso	0.6935	**0.7572**	0.4692	0.7846	-6.5%	**-1.9%**	14.2%	36.8%
UCAIR	**0.7272**	**0.7721**	**0.4287**	0.6428	**-1.9%**	**0.0%**	**4.4%**	12.1%

Table 19. Comparison of each method's performance with the top achiever for every measure (micro-averaged results including the *"AOL 2006"* sample).

	$F_{\beta}.\ \beta = 1$	$F_{\beta}.\ \beta = 1.5$	ERR	SER	$\Delta F_{\beta}.\ \beta = 1$	$\Delta F_{\beta}.\ \beta = 1.5$	ΔERR	ΔSER
Buzikashvili	**0.7694**	**0.7996**	0.3747	0.5113	**-3.6%**	**-2.8%**	11.5%	16.6%
CondProbs (5 mins)	0.7002	0.6729	0.4613	0.5423	-12.3%	-18.2%	37.3%	23.7%
DynSlidingWindow	0.7047	0.6836	0.4559	0.5465	-11.7%	-16.9%	35.6%	24.6%
Geometric	**0.7980**	**0.8229**	**0.3361**	**0.4385**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
QueryContent	**0.7639**	**0.8091**	0.3820	0.5524	**-4.3%**	**-1.7%**	13.7%	26.0%
QueryContentExpanded	0.7521	0.7785	0.3974	0.5439	-5.8%	-5.4%	18.2%	24.0%
SecoCardoso	0.7448	**0.8043**	0.4066	0.6319	-6.7%	**-2.3%**	21.0%	44.1%
UCAIR	**0.7754**	**0.8145**	0.3669	0.5134	**-2.8%**	**-1.0%**	9.2%	17.1%

Table 20. Comparison of each method's performance with the top achiever for every measure (macro-averaged results excluding the *"AOL 2006"* sample).

	$F_{\beta}.\ \beta = 1$	$F_{\beta}.\ \beta = 1.5$	ERR	SER	$\Delta F_{\beta}.\ \beta = 1$	$\Delta F_{\beta}.\ \beta = 1.5$	ΔERR	ΔSER
Buzikashvili	**0.6844**	**0.7211**	0.4798	0.7276	**-3.8%**	**-3.4%**	7.1%	11.6%
CondProbs (5 mins)	0.5614	0.5295	0.6098	0.7585	-21.1%	-29.0%	36.1%	16.3%
DynSlidingWindow	0.5664	0.5384	0.6050	0.7641	-20.4%	-27.8%	35.1%	17.2%
Geometric	**0.7114**	**0.7443**	**0.4479**	**0.6522**	**0.0%**	**-0.2%**	**0.0%**	**0.0%**
QueryContent	**0.6786**	**0.7354**	0.4865	0.8043	**-4.6%**	**-1.4%**	8.6%	23.3%
QueryContentExpanded	**0.6997**	**0.7446**	**0.4618**	0.7120	**-1.6%**	**-0.2%**	**3.1%**	9.2%
SecoCardoso	0.6604	**0.7286**	0.5071	0.8979	-7.2%	**-2.3%**	13.2%	37.7%
UCAIR	**0.6975**	**0.7461**	**0.4645**	0.7285	**-2.0%**	**0.0%**	**3.7%**	11.7%

Table 21. Comparison of each method's performance with the top achiever for every measure (macro-averaged results including the *"AOL 2006"* sample).

	$F_{\beta}.\ \beta = 1$	$F_{\beta}.\ \beta = 1.5$	ERR	SER	$\Delta F_{\beta}.\ \beta = 1$	$\Delta F_{\beta}.\ \beta = 1.5$	ΔERR	ΔSER
Buzikashvili	**0.7126**	**0.7471**	0.4464	0.6531	**-3.6%**	**-2.9%**	8.0%	12.5%
CondProbs (5 mins)	0.6039	0.5740	0.5675	0.6978	-18.3%	-25.4%	37.3%	20.2%
DynSlidingWindow	0.6097	0.5844	0.5614	0.7014	-17.5%	-24.1%	35.8%	20.8%
Geometric	**0.7394**	**0.7697**	**0.4134**	**0.5806**	**0.0%**	**0.0%**	**0.0%**	**0.0%**
QueryContent	**0.7075**	**0.7600**	0.4526	0.7131	**-4.3%**	**-1.3%**	9.5%	22.8%
QueryContentExpanded	**0.7256**	**0.7669**	0.4306	0.6382	**-1.9%**	**-0.4%**	4.2%	9.9%
SecoCardoso	0.6876	**0.7531**	0.4760	0.8073	-7.0%	**-2.2%**	15.1%	39.0%
UCAIR	**0.7227**	**0.7681**	0.4342	0.6551	**-2.3%**	**-0.2%**	5.0%	12.8%

7. Implications and conclusion

Because of the pervasive presence of search engines in users' Web interactions query logs raise many privacy and ethical concerns even if just used for academic purposes. It seems that simple "prophylactic" measures –such as segmenting records of users' queries into shorter segments– would dispel many of the concerns on this matter while preserving most of the usefulness of the data for researchers. In this regard, the author maintains that topical session detection methods could allow the collection of query logs spanning several days, even months, while reducing to a minimum the risk of privacy leaks.

Thus, this study contributes to our understanding of this issue in several important ways. First, it provides a thorough review of the state of the art with regards to the session detection problem. Second, it proposes an evaluation framework for such sessionization methods adapting two performance measures (namely, *ERR* and *SER*). Third, it describes a new test collection which is, to the best of our knowledge, the largest and most exhaustive manually segmented query log to the date, comprising about 95,000 queries segmented into 34,530 topical sessions. Fourth, this author described a new heuristic-based session detection method able to operate in real time on query streams. Fifth, this study has shown that most of the session detection methods outperform the commonly used baseline and that the new method proposed by this author consistently outperforms all the other described techniques.

This study also has limitations. First, the way in which queries issued by robots were removed could be greatly improved, provided an accurate method to tell apart humans from software agents. Second, machine-learning methods were evaluated with the parameterizations provided by their original authors and not trained on a subset of the segmented query log. Third, manually segmenting the original datasets was a daunting task and, thus, future work in this area should study the possibility of applying the pooling method to the evaluation of session detection techniques.

Hence, to achieve the long term goal of providing academia with query logs not tantalized by privacy concerns, further research is needed in the following lines: (1) accurate ways to automatically detect software agents querying search engines; (2) machine-learning methods of session detection; and (3) development of datasets, tools and infrastructures for the evaluation and development of session detection methods.

8. Acknowledgements

The author would like to thank Dr. Jim Jansen for providing access to most of the query logs employed in this research. He is also grateful to Professor Asunción Lubiano-Gómez for many discussions on statistical aspects and to the anonymous reviewers for their valuable comments and advice. This work was partially financed by grant UNOV-08-MB-13 from the University of Oviedo and completed while the author was visiting the University of Plymouth (UK).

9. References

[1] E. Adar, "User 4XXXXX9: Anonymizing Query Logs," *Query Log Analysis: Social and Technological Challenges*, Workshop in WWW, vol. 7, 2007. Available at: http://www2007.org/workshops/paper_52.pdf (accessed 24 November 2008)

[2] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," *Proc. of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 19-26.

[3] N. Anderson, "The ethics of using AOL search data," *Ars Technica*, 2006. Available at: http://arstechnica.com/news.ars/post/20060823-7578.html (accessed 16 July 2008)

[4] R. Baeza-Yates, "Web Usage Mining in Search Engines," in: A. Scime (Editor), *Web Mining: Applications and Techniques*, Idea Group, 2005, pp. 307-321.

[5] M. Barbaro and T. Zeller, "A Face Is Exposed for AOL Searcher No. 4417749," *New York Times*, vol. 9, 2006. Available at: http://www.nytimes.com/2006/08/09/technology/09aol.html (accessed 24 November 2008)

[6] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," *Proc. of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 407-416.

[7] N. Buzikashvili, "An exploratory web log study of multitasking," *Proc. of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 623-624.

[8] N. Buzikashvili, "Automatic Task Detection in the Web Logs and Analysis of Multitasking," *Lecture Notes in Computer Science*, vol. 4312, 2006, pp. 131-140.

[9] N. Buzikashvili, "Sliding window technique for the web log analysis," *Proc. of the 16th international conference on World Wide Web*, 2007, pp. 1213-1214.

[10] N. Buzikashvili and B.J. Jansen, "Limits of the Web log analysis artifacts," *Workshop on logging traces of Web activity, WWW 2006*, 2006. Available at: http://torch.cs.dal.ca/~www2006/buzik-www2006-MechanicsDataCollection.pdf (accessed 24 November 2008)

[11] W.B. Cavnar and J.M. Trenkle, "N-Gram-Based Text Categorization," *Proc. of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161-175.

[12] S. Chien and N. Immorlica, "Semantic similarity between search engine queries using temporal correlation," *Proc. of the 14th international conference on World Wide Web*, 2005, pp. 2-11.

[13] N. Chinchor and G. Dungca, "Four scorers and seven years ago: the scoring method for MUC-6," *Proc. of the 6th conference on Message Understanding*, 1995, pp. 33-38.

[14] S.L. Chuang and L.F. Chien, "Enriching Web taxonomies through subject categorization of query terms from search engine logs," *Decision Support Systems*, vol. 35, 2003, pp. 113-127.

[15] S.L. Chuang and L.F. Chien, "A practical web-based approach to generating topic hierarchy for text segments," *Proc. of the 13th ACM conference on Inf. and Knowledge management*, 2004, pp. 127-136.

[16] S. Cucerzan and E. Brill, "Spelling correction as an iterative process that exploits the collective knowledge of web users," *Proc. of EMNLP*, vol. 4, 2004, pp. 293-300.

[17] H. Cui *et al.*, "Probabilistic query expansion using query logs," *Proc. of the 11th international conference on World Wide Web*, 2002, pp. 325-332.

[18] H. Cui *et al.*, "Query expansion by mining user logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, 2003, pp. 829-839.

[19] D. Downey, S. Dumais, and E. Horvitz, "Models of Searching and Browsing: Languages, Studies, and Applications," *Proc. of IJCAI*, 2007, pp. 1465-1472.

[20] T.R. Girill, "Online access AIDS for documentation: a bibliographic outline," *ACM SIGIR Forum*, vol. 18, 1985, pp. 24-27.

[21] A. Göker and D. He, "Analysing Web Search Logs to Determine Session Boundaries for User-Oriented Learning," *Proc. of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2000, pp. 319-322.

[22] K. Hafner, "Researchers Yearn to Use AOL Logs, but They Hesitate," *New York Times*, vol. 23, 2006. Available at: http://www.nytimes.com/2006/08/23/technology/23search.html (accessed 24 November 2008)

[23] M.H. Hansen and E. Shriver, "Using navigation data to improve IR functions in the context of web search," *Proc. of the 10th international conference on Information and knowledge management*, 2001, pp. 135-142.

[24] D. He and A. Göker, "Detecting session boundaries from Web user logs," *Proc. of the 22nd annual colloquium on information retrieval research*, Cambridge, April, 2000, pp. 57-66.

[25] D. He, A. Göker, and D.J. Harper, "Combining evidence for automatic Web session identification," *Information Processing and Management*, vol. 38, 2002, pp. 727-742.

[26] C.K. Huang, L.F. Chien, and Y.J. Oyang, “Relevant term suggestion in interactive web search based on contextual information in query session logs,” *Journal of the American Society for Information Science and Technology*, vol. 54, 2003, pp. 638-649.

[27] B.J. Jansen and A. Spink, “Methodological approach in discovering user search patterns through Web log analysis,” *ACM SIGIR Forum*, vol. 32, 2000, pp. 5-17.

[28] B.J. Jansen and A. Spink, “An Analysis of Web Documents Retrieved and Viewed,” *Proc. of the 4th International Conference on Internet Computing*, 2003, pp. 65-69.

[29] B.J. Jansen and A. Spink, “An analysis of Web searching by European AlltheWeb. com users,” *Information Processing and Management*, vol. 41, 2005, pp. 361-381.

[30] B.J. Jansen and A. Spink, “How are we searching the World Wide Web? A comparison of nine search engine transaction logs,” *Information Processing and Management*, vol. 42, 2006, pp. 248-263.

[31] B.J. Jansen, A. Spink, and J. Pedersen, “A temporal comparison of AltaVista Web searching,” *Journal of the American Society for Information Science and Technology*, vol. 56, 2005, pp. 559-570.

[32] B.J. Jansen, A. Spink, and T. Saracevic, “Real life, real users, and real needs: a study and analysis of user queries on the web,” *Information Processing and Management*, vol. 36, 2000, pp. 207-227.

[33] B.J. Jansen *et al.*, “Real life information retrieval: a study of user queries on the Web,” *ACM SIGIR Forum*, vol. 32, 1998, pp. 5-17.

[34] B.J. Jansen et al., “Automated gathering of Web information: An in-depth examination of agents interacting with search engines,” *ACM Transactions on Internet Technology (TOIT)*, vol. 6, 2006, pp. 442-464.

[35] B.J. Jansen *et al.*, “Defining a Session on Web Search Engines,” *Journal of the American Society for Information Science and Technology*, vol. 58, 2007, pp. 862-871.

[36] T. Joachims, “Optimizing search engines using clickthrough data,” *Proc. of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133-142.

[37] T. Joachims *et al.*, “Accurately interpreting clickthrough data as implicit feedback,” *Proc. of the 28th ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 154-161.

[38] R. Jones *et al.*, “Generating query substitutions,” *Proc. of the 15th international conference on World Wide Web*, 2006, pp. 387-396.

[39] D. Kristol and L. Montulli, *HTTP State Management Mechanism*, RFC 2109, February 1997. Available at: http://www.ietf.org/rfc/rfc2965.txt (accessed 24 November 2008)

[40] T. Lau and E. Horvitz, “Patterns of Search: Analyzing and Modeling Web Query Refinement,” *Proc. of the Seventh International Conference on User Modeling*, 1999, pp. 119-128.

[41] D.D. Lewis, “Evaluating text categorization,” *Proc. of Speech and Natural Language Workshop*, 1991, pp. 312-318.

[42] J. Makhoul *et al.*, “Performance measures for information extraction,” *Proc. of the DARPA Broadcast News Workshop*, 1999, pp. 249-252.

[43] E.P. Markatos, “On caching search engine query results,” *Computer Communications*, vol. 24, 2001, pp. 137-143.

[44] M.E. Maron and J.L. Kuhns, “On Relevance, Probabilistic Indexing and Information Retrieval,” *Journal of the ACM*, vol. 7, 1960, pp. 216-244.

[45] D. Metzler, S. Dumais, and C. Meek, “Similarity Measures for Short Segments of Text,” *Lecture Notes in Computer Science*, vol. 4425, 2007, pp. 16-27.

[46] Microsoft, *Microsoft Research Microsoft Live Labs: Accelerating Search in Academic Research 2006, Request for Proposals (2006).* Available at: http://research.microsoft.com/ur/us/fundingopps/RFPs/Search_2006_RFP.aspx (accessed 16 July 2008).

[47] G.C. Murray, J. Lin, and A. Chowdhury, “Identification of User Sessions with Hierarchical Agglomerative Clustering,” *ASIS&T*, vol. 6, 2006, pp. 3-8.

[48] S. Özmutlu, “Automatic new topic identification using multiple linear regression,” *Information Processing and Management*, vol. 42, 2006, pp. 934-950.

[49] S. Özmutlu and B. Buyuk, “Using Monte-Carlo simulation for automatic new topic identification of search engine transaction logs,” *Proc. of the 39th conference on Winter simulation*, 2007, pp. 2306-2314.

[50] H.C. Özmutlu and F. Çavdur, “Application of automatic topic identification on excite web search engine data logs,” *Information Processing and Management*, vol. 41, 2005, pp. 1243-1262.

[51] S. Özmutlu and F. Çavdur, “Neural network applications for automatic new topic identification,” *Online Information Review*, vol. 29, 2005, pp. 34-53.

[52] H.C. Özmutlu, F. Çavdur, and S. Özmutlu, “Cross-validation of neural network applications for automatic new topic identification,” *Journal of the American Society for Information Science and Technology*, vol. 59, 2008, pp. 339-362.

[53] H.C. Özmutlu, F. Çavdur, and S. Özmutlu, “Automatic new topic identification in search engine transaction logs,” *Internet Research*, vol. 16, 2006, pp. 323-338.

[54] S. Özmutlu, H.C. Özmutlu, and B. Buyuk, “Using conditional probabilities for automatic new topic identification,” *Online Information Review*, vol. 31, 2007, pp. 491-515.

[55] S. Özmutlu, H.C. Özmutlu, and A. Spink, “Automatic New Topic Identification in Search Engine Transaction Logs Using Multiple Linear Regression,” *Proc. of the 41st Annual Hawaii International Conference on System Sciences*, 2008, pp. 140-140.

[56] G. Pass, A. Chowdhury, and C. Torgeson, “A picture of search,” *Proc. of the 1st international conference on Scalable information systems*, 2006. Available at: http://doi.acm.org/10.1145/1146847.1146848 (accessed 24 November 2008)

[57] F. Radlinski and T. Joachims, “Query chains: learning to rank from implicit feedback,” *Conference on Knowledge Discovery in Data*, 2005, pp. 239-248.

[58] C.J. Van Rijsbergen, “Evaluation”, in *Information Retrieval*, Butterworth-Heinemann, Newton, MA, USA, 1979, pp. 112-140.

[59] D.E. Rose and D. Levinson, “Understanding user goals in web search,” *Proc. of the 13th international conference on World Wide Web*, 2004, pp. 13-19.

[60] M. Sahami and T.D. Heilman, “A web-based kernel function for measuring the similarity of short text snippets,” *Proc. of the 15th international conference on World Wide Web*, 2006, pp. 377-386.

[61] N. Seco and N. Cardoso, *Detecting User Sessions in the Tumba! Query Log* (Unpublished manuscript, Department of Informatics Engineering, University of Coimbra, Portugal, 2006). Available at: http://eden.dei.uc.pt/~nseco/tumba.pdf (accessed 24 November 2008)

[62] X. Shen, B. Tan, and C. Zhai, “Implicit user modeling for personalized search,” *Proc. of CIKM*, 2005, pp. 824-831.

[63] X. Shi and C.C. Yang, “Mining related queries from search engine query logs,” *Proc. of the 15th international conference on World Wide Web*, 2006, pp. 943-944.

[64] C. Silverstein *et al.*, “Analysis of a very large web search engine query log,” *ACM SIGIR Forum*, vol. 33, 1999, pp. 6-12.

[65] A. Spink, B.J. Jansen, and H.C. Özmutlu, “Use of query reformulation and relevance feedback by Excite users,” *Internet Research: Electronic Networking Applications and Policy*, vol. 10, 2000, pp. 317-328.

[66] A. Spink, H.C. Özmutlu, and S. Özmutlu, “Multitasking information seeking and searching processes,” *Journal of the American Society for Information Science and Technology*, vol. 53, 2002, pp. 639-652.

[67] A. Spink *et al.*, “Modeling Users' Successive Searches in Digital Environments,” *D-Lib Magazine*, 1998. Available at: http://www.dlib.org/dlib/april98/04spink.html (accessed 24 November 2008)

[68] A. Spink *et al.*, “Searching the web: The public and their queries,” *Journal of the American Society for Information Science and Technology*, vol. 52, 2001, pp. 226-234.

[69] A. Spink *et al.*, “From E-Sex to E-Commerce: Web Search Changes,” *Computer*, vol. 35, 2002, pp. 107-109.

[70] A. Spink *et al.*, “US versus European web searching trends,” *ACM SIGIR Forum*, vol. 36, 2002, pp. 32-38.

[71] A. Spink *et al.*, “Multitasking during Web search sessions,” *Information Processing and Management*, vol. 42, 2006, pp. 264-275.

[72] R. Sproat and T. Emerson, “The First International Chinese Word Segmentation Bakeoff,” *Proc. of the Second SIGHAN Workshop on Chinese Language Processing*, 2003, pp. 133-143.

[73] J.T. Sun *et al.*, “Web-page summarization using clickthrough data,” *Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 194-201.

[74] D.R. Swanson, “Information retrieval as a trial-and-error process,” *Library Quarterly*, vol. 47, 1977, pp. 128-148.

[75] J.R. Wen, J.Y. Nie, and H.J. Zhang, “Query clustering using user logs,” *ACM Transactions on Information Systems*, vol. 20, 2002, pp. 59-81.

[76] J.R. Wen and H.J. Zhang, “Query Clustering in the Web Context,” in Wu, Xiong and Shekhar (Eds.), *Information Retrieval and Clustering*, pp. 195-226, (Kluwer Academic Publishers, 2003).

[77] J.R. Wen *et al.*, “Clustering user queries of a search engine,” *Proc. of the 10th international conference on World Wide Web*, 2001, pp. 162-168.

[78] D. Wolfram, “A Query-Level Examination of End User Searching Behaviour on the Excite Search Engine,” *Proc. of ACSI 2000*, 2000. Available at: http://www.cais-acsi.ca/proceedings/2000/wolfram_2000.pdf (accessed 24 November 2008)

[79] D. Wolfram *et al.*, “Vox Populi: The Public Searching of the Web,” *Journal of the American Society for Information Science and Technology*, vol. 52, 2001, pp. 1073-1074.

[80] Y. Xie and D. O’Hallaron, “Locality in Search Engine Queries and Its Implications for Caching,” *IEEE INFOCOM*, 2002. Available at: http://www.comsoc.org/confs/ieee-infocom/2002/papers/307.pdf (accessed 24 November 2008)

[81] L. Xiong and E. Agichtein, “Towards Privacy-Preserving Query Log Publishing,” in Aimtay, Murray and Teevan (Eds.), *Query Log Analysis: Social and Technological Challenges. A workshop at the 16th International World Wide Web Conference (WWW 2007)*, 2007. Available at: http://www2007.org/workshops/paper_136.pdf (accessed 24 November 2008)

[82] G.R. Xue *et al.*, “Optimizing web search using web click-through data,” *Proc. of the 13th ACM conference on Information and knowledge management*, 2004, pp. 118-126.

[83] D. Zhang and Y. Dong, “A novel Web usage mining approach for search engines,” *Computer Networks*, vol. 39, 2002, pp. 303-310.

[84] Y. Zhang and A. Moffat, “Some Observations on User Search Behavior,” *Proc. of the 11th Australasian Document Computing Symposium*, 2006, pp. 1-8.

[85] Y. Zhang and A. Moffat, “Separating Human and Non-Human Web Queries,” *Proc. of the Web Information Seeking and Interaction Workshop*, Amsterdam, 2007, pp. 13-16.