Not gone, but forgotten: Helping users re-find web pages by identifying those which are most likely to be lost

Karl Gyllstrom
Department of Computer Science
Katholieke Universiteit Leuven
Leuven, Belgium
karl.gyllstrom@cs.kuleuven.be

Elin Rønby Pedersen
Google, Inc.
Mountain View, CA
USA
elinp@google.com

ABSTRACT

We describe LostRank, a project in its formative stage which aims to produce a way to rank results in re-finding search engines according to the likelihood of their being lost to the user. To this end, we have explored a number of ideas, including applying users' temporal document access patterns to determine the documents that are both important and have not been recently accessed (indicating greater potential for loss), understanding users' topical access patterns to determine the topics that are more unfamiliar and hence more difficult to re-find documents within, and assessing users' difficulties in originally finding documents in order to predict future difficulties in re-finding them. As a position paper, we use this as an opportunity to describe early work, invite collaboration with others, and further the case for the use of temporal access patterns as a source for assisting users' re-finding of personal documents.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]
General Terms: Human Factors
Keywords: Re-finding, ranking, log analysis

1. INTRODUCTION

Personal document collections grow constantly. Each day we access a significant number of new web pages, many of which we will probably never access again. One challenge is that finding a document within a large collection requires a specific query to distinguish the file from others in the collection. As time passes, our recollection of document specifics – with which we would formulate queries – decays. In other words, as time goes on, not only does our document collection grow larger – and hence harder to search – but our ability to issue good queries declines.

One area that deserves attention is the ranking function for search results, as a strong one can allow desktop search to produce good results for vague queries on large personal datasets. Additionally, it allows for a more aggressive expansion of users' queries to include topical or syntactic synonyms, as users are more likely to forget key terms (or use wrong terms) when re-finding documents accessed further into the past. Ranking is an important subject in re-finding because it addresses a fundamentally different problem than search for new information, and limits what can be imported from mature domains such as web search. For example, on the web, algorithms like PageRank or HITS are effective ranking functions because they promote credible or authoritative pages, and when users seek answers to new questions, they desire answers that are most likely to be accurate. Credibility is defined by lots of incoming links from other credible pages, which has the effect of more highly ranking pages that are considered more important by the large community of web authors.

We argue that this is the opposite of what is desired for re-finding tasks. Though hyperlink structure is not present on users' filesystems, consider an analog assessment of importance, such as number of shortcuts, or proximity to the home or desktop directories. These qualities indicate that a document is quite important, and yet, they provide evidence that the document is unlikely to be lost, as it is readily accessible. Lost information tends to be that which is hidden within a difficult navigation path, such as within deeply nested directories or large files.

In our work, we have pursued ways to rank documents according to their likelihood of being "lost". In this context, we define lost documents to be those which a user has previously accessed, desires to access again, and is unable to find using traditional search methods, such as text-based desktop search. Classifying a document as lost is obviously a difficult and large endeavor. We have developed a few ideas which we are beginning to explore and evaluate, including page access patterns, topic access patterns, and difficulties surrounding the original document discovery. We describe these below, but first, let us summarize our use of personal web log data.

2. WEB LOG DATA AND ABSTRACTIONS

Our goal is to break a user's document activity into higher-level abstractions that allow us to better reason about it. In this work, we focus on web history because it is easy to extract (e.g., via Firefox), and, since it contains queries, it allows us to better understand information seeking behavior. We believe this approach could be extended to general document activity recording systems.

A web history is a time-ordered sequence of events, where an event is either a query, including query text, or a page click, including the URL and page contents. We process it using a two-fold approach: First, the history is separated into segments, where segments encompass a sequence of queries and page visits that occur within 5 minutes of each other. A segment roughly (though imperfectly) approximates a single task (e.g., searching for housing). Second,

Copyright is held by the author/owner(s). SIGIR'10, Workshop on Desktop Search, July 23, 2010, Geneva, Switzerland

the LDA topic detection algorithm [1] is run on the contents of the pages within these segments. These two approaches assign to each page a set of tasks and topics – including the relative strength of relationship between the page and each of its topics [2].

For each segment, we assign a difficulty assessment, which is measurement of the apparent difficulty of the information seeking task. We have selected a number of qualities, including number of queries, number of query reformulations (modifications of unsuccessful search attempts), length of session, number of queries for which no results are clicked (indicating poor queries), and average page view time. Pages within a segment inherit its difficulty score.

3. RANKING COMPONENTS

In this section we describe a few ranking components we have explored. After independently evaluating them we hope to combine them into a comprehensive ranking function. We envision adding more as this project matures.

3.1 Access patterns

As memory decays with time, the likelihood of a document being lost increases with the time since its last access. However, time-of-last-access alone is not sufficient to suitably rank documents. We use look beyond time-of-last-access to consider larger access patterns. For example, consider two pages that were last accessed by a user one month ago. Constrained to time-of-last-access, we would rank these pages as equally lost. Let us assume that one of the pages was first viewed at this time, while the other page has been accessed once per month for the last 2 years. We might reason that the latter page is less likely to be lost because its time-of-last-access is consistent with a larger pattern of access, and assign it a weaker rank.

Another case is we consider is when documents' access patterns change. For example, a page that was very frequently accessed for a period of several months, but then not accessed at all for a year, has a pattern that we refer to as *dormant*. This pattern fits our definition of lost in that it indicates that the page was once important to the user (indicating that they may want to eventually use it again), and that the user's familiarity with the page has declined (as evidenced by not being accessed for a long time).

3.2 Topic patterns

We extend the above notion to include topic, with the observation that users' revisitation patterns vary according to topic. For example, queries for code documentation might frequently be navigation-style queries for which the user has little difficulty finding relevant answers (e.g., looking up the Java `Set` class). Other topics, such as health, may involve more complex search processes where the answer to a question is more vague.

Our current implementation is to determine, for each page, the most closely linked LDA topics, and record an event for the topic at that point. This allows us to build an access pattern for each topic, and associate topical activity to each page. Pages with more dormant topics may be those which are more likely to be lost. The advantage of using topic is that we can reason about pages that the user has not accessed enough for a reliable pattern to emerge (e.g., the user only looks up code for Java class `String` once, but looks up code for Java classes routinely; it would therefore be assigned a weaker rank as the topic pattern indicates it is likely easily re-found).

3.3 Difficulty before original access

We consider the path a user takes to originally access a page, using the difficulty assessment described in Section 2. Repeated navigational queries – web queries that are intended to find a specific page (e.g., "ebay") – suggest an easily re-found page. Pages discovered after long trails of queries and query reformulations indicate that the overarching task may have been more vague, or that the user lacked prior knowledge before the research task. We hypothesize that, as the latter are cases where the user's understanding of the topic is weaker, the user's recollection of terms from pages from difficult tasks will be worse, especially for the pages accessed later in the task. For example, a research path that began on energy-efficient buildings may have resulted in research on passive windows, the latter being a term less easily remembered if the user continues or restarts the research weeks or months later. Their query formulation may tend toward their original terminology rather than the terminology used in pages accessed after the task evolved.

4. CONCLUDING REMARKS

Most of the ideas described in this work originated from observations on a small number of very large query logs that volunteers offered for our use. We would like to evaluate them directly on a larger pool of user data, and invite the comments and participation of the community. In particular, we would like to see more research emphasis on personalized ranking in the context of re-finding.

There are a number of related works that have inspired this work. Several systems aim to improve document refinding by tracing users' desktop activity, for example, by detecting task relationships [3, 4]; our work would benefit from these systems' tracing approaches, and allow us to integrate better task representations. The Re:search engine enhances web search by integrating previously accessed pages into search results for queries with similarity to previously issued queries [5]; we share a common goal, although we focus on determining which previously accessed pages to show to users.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] E. R. Pedersen, K. Gyllstrom, S. Gu, and P. J. Hong. Automatic generation of research trails in web history. In *IUI '10*, pages 369–372, New York, NY, USA, 2010. ACM.

[3] T. Rattenbury and J. Canny. CAAD: an automatic task support system. In *CHI '07*, pages 687–696, New York, NY, USA, 2007. ACM.

[4] C. A. N. Soules and G. R. Ganger. Connections: using context to enhance file search. *SIGOPS Oper. Syst. Rev.*, 39(5):119–132, 2005.

[5] J. Teevan. The re:search engine: simultaneous support for finding and re-finding. In *UIST '07*, pages 23–32, New York, NY, USA, 2007. ACM.