B-hist: Entity-Centric Search over Personal Web Browsing History

Michele Catasta\textsuperscript{a,*}, Alberto Tonon\textsuperscript{b}, Gianluca Demartini\textsuperscript{b}, Jean-Eudes Ranvier\textsuperscript{a}, Karl Aberer\textsuperscript{a}, Philippe Cudré-Mauroux\textsuperscript{b}

\textsuperscript{a}EPFL—Switzerland
\textsuperscript{b}eXascale Infolab, University of Fribourg—Switzerland

Abstract

Web Search is increasingly entity-centric; as a large fraction of common queries target specific entities, search results get progressively augmented with semi-structured and multimedia information about those entities. However, search over personal Web browsing history still revolves around keyword-search mostly. In this paper, we present a novel approach to answer queries over Web browsing logs that takes into account entities appearing in the Web pages, user activities, as well as temporal information. Our system, B-hist, aims at providing Web users with an effective tool for searching and accessing information they previously looked up on the Web by supporting multiple ways of filtering results using clustering and entity-centric search. In the following, we present our system and motivate our User Interface (UI) design choices by detailing the results of a survey on Web browsing and history search. In addition, we present an empirical evaluation of our entity-based approach used to cluster Web pages.

© 2014 Published by Elsevier Ltd.

Keywords: browsing history, personal memories, entity-centric search

1. Introduction

Searching over one’s own personal browsing history is useful to locate information that was previously seen and that is once again needed. Often, when trying to remember some previously looked-up information, people rather search the Web instead of searching over locally stored files or over their Web browsing history. This is mostly due to the fact that search tools for the Web are more effective than those available for desktop or browsing history search.

Web Search today exploits semantic data in order to improve search results and provide more information satisfying the user’s query intent: Search Engine Result Pages (SERPs) are enriched with structured content including pictures, maps, factual data—in addition to the standard links pointing to Web pages [1, 2]. This is possible thanks to structured knowledge bases and Linked Open Data (LOD) datasets such as Freebase [3] and thanks to semantic annotations of Web pages using, for instance, schema.org. Search over personal Web browsing logs is a related task, though it has in our opinion not yet received the full benefits of semantic techniques. While studies have been shown that re-finding information on-line is more effective when supported by search tools [4], most browsers provide a very limited keyword-based search over previously visited pages, which has not changed much in the last 20 years.

*Corresponding author

Email addresses: michele.catasta@epfl.com (Michele Catasta), alberto.tonon@unifr.ch (Alberto Tonon), gianluca.demartini@unifr.ch (Gianluca Demartini), jean-eudes.ranvier@epfl.ch (Jean-Eudes Ranvier), karl.aberer@epfl.com (Karl Aberer), pcm@unifr.ch (Philippe Cudré-Mauroux)
In this paper, we present a new system that lets users search over their personal Web browsing history in an entity-centric fashion. The goal of our system, called B-hist (standing for 'Better history'), is to bring entity-centric access to personal browsing activities thanks to semantic technologies such as the ones we developed in our recent pieces of work [5, 6] for entity disambiguation and entity type selection.

Semantic technologies leveraged by B-hist include the use of structured metadata about entities extracted from web pages which have been visited by the users: our system leverages entities from DBpedia and create links from web pages to entities. Moreover, B-hist leverages a broad entity type hierarchy build on top of DBpedia, YAGO, and schema.org types.

By mining entities in Web pages and leveraging their types to cluster pages in meaningful groups, we allow users to access their Web history from multiple entry points: They can type queries which get auto-completed with the entities mentioned in their history. They also can filter results based on the time dimension thanks to a heat map calendar showing browsing activity over time, and by clicking on entity types or on clusters of coherent Web browsing sessions.

The rest of the paper is structured as follows: Section 2 briefly summarizes work from related areas and existing software aiming at enhancing the Web history search experience. Section 3 presents the different components of B-hist. Section 4 describes the results of an online survey on Web browsing and history search based on more than 200 participants. It also offers the results of our evaluation of different approaches for clustering Web pages. Finally, Section 5 concludes the paper and highlights the main novelties of our system.

2. Related Work

2.1. Web Search

Web Search has been studied extensively over the last 15 years. Early work in the area includes the study of different types of Web search information needs: informational, navigational, and transactional [7]. Notable focuses in this domain have been put on both improving efficiency of search engines by means of mining user activities [8] and on proposing effective information retrieval models [9].

More recently, Semantic Web technologies have been used to enhance the experience of Web search users. Examples include the enrichment of search engine result pages with maps, images, and vertical search results (e.g., news, blogs) intermixed with organic search results coming from an inverted index built on top of a Web crawl [1]. For few years, a background knowledge graph is used by search engines in order to provide rich SERPs for entity-centric queries [2] as it has been shown that many Web search queries are indeed about entities [10].

In our work we aim at applying state-of-the-art Semantic Web techniques that have been applied to Web search also to the problem of Web Browsing History Search.

2.2. Web Browsing History Search

In [11, 12] Cockburn and McKenzie show that 81% of the pages browsed by their sample of Web-users were actually re-visits of some page previously visited. This provides a good motivation for research on Web browsing history. In particular, in [13] Mayer and Bederson present a system that allows the users to organize their browsing history in sessions, where a session is defined as a “meaningful unit in which somebody uses the Web with a more or less specific goal in mind”. The user of the system has to manually select the session he/she is working on (e.g., “organizing a conference”). The goal of B-hist is to automatically detect sessions and create corresponding clusters of Web-pages. While Mayer and Bederson’s system uses graphs representing connections among pages to represent a session, in our User Interface (UI) we use a more user-friendly representation consisting of pictures of the main entities appearing in the session.

In [14] the authors describe History-Centric Browsing (HCB): A system that displays information from Web browsing history pages which are related to the Web page the user is currently watching. In HCB, two pages can be related because they are visited one directly after the other, because their content is similar, or because they are two different versions of a page with the same URL. We note that HCB does not take into consideration entities and their semantic relations in order to organize the user’s browsing history.

A more recent study [15] showed that navigation strategies vary drastically based on user habits. This is also confirmed by our survey, which shows that some features are more useful than others depending on how often people use Web history search functionalities.
2.3. Related Products

More recently, the Mozilla foundation has been working on a related system called Pancake\(^1\) whose goal is to integrate search results from browsing history, social streams, and Web search. While its focus is on integrating content from different sources, the system we propose rather aims at semantically enriching the search experience over personal browsing history easing the access and recall of previously seen information.

A commercial product related to B-hist is being developed by CottonTracks\(^2\) and provides a clustered access to personal browsing history. However, B-hist provides a much richer set of information access functionalities thanks to the semantic enrichment of one’s Web browsing history, which is its core competitive advantage.

Additionally to CottonTracks our system B-hist also uses other semantic dimensions to enable re-finding of information: Entities are displayed to users to help them recall what they have encountered in the past; Our clustering approach is session-based and not just based on single page clustering as in CottonTracks; Finally, our system also allows for time-range search.

3. System Description

Our system provides a multi-dimensional access to one’s personal Web history by letting users select the desired pieces of information by means of several filters: temporal, entity-centric, and session-based. In the following, we describe the main components of B-hist and its data processing back-end architecture.

3.1. Chrome Browser Extension

The initial data collection is handled by a Web browser extension, which is responsible both to gather raw data from the user browsing activities as well as to let the user set preferences and to access the search dashboard of B-hist. Specifically, the settings of the extension allow the user to filter-out
some domains as well as to allow/disallow https domains from being stored, indexed, and searched by the system. The extension also opens a new browser tab displaying the welcome screen of B-hist where the users can start looking for information in their browsing history (see Figure 1).

At the moment of writing this paper (June 2014), Chrome is the browser of choice for 59% of the Web users, therefore we initially focused on developing an extension for the Chrome platform. Considering the system architecture of B-hist, though, the extension could be easily ported to other browsers (e.g., Firefox, Safari, etc.) as we leverage only standardized, multi-platform features.

3.2. B-hist Data Processing

Once the raw HTML data is gathered from the page that is accessed, it goes through the B-hist back-end, which is responsible to run our TRank [6] processing pipeline (see Figure 2), where the main textual content is kept (using approaches from [16]), entities are extracted (using a Conditional Random Field approach trained on a news corpus [17]), and entity types get selected (using approaches from [6]). The resulting metadata are stored and indexed in B-hist (see Figure 2).

To store and index data (e.g., timestamps and cluster information) both an inverted index (i.e., Apache Lucene\(^\text{3}\)) and a lightweight DBMS (i.e., SQLite) are used. Those indexes are also exploited in order to compute the list of words the user can auto-complete (by using the suggestSimilar feature provided by Lucene) The raw HTML coming from the browser extension is however not stored in B-hist, as this would require too much storage on the long run.

In parallel to the TRank pipeline, a batch process of session discovery and categorization is accessing the data from the browser and creating additional metadata in order to regroup pages in coherent sessions with a common user intent.

\(^\text{3}\)http://lucene.apache.org/
Clustering of Web Pages. Each Web page $p$ is identified by the list $\tau(p)$ of the top-$n$ most frequent entity types associated to the entities it contains. In order to do this, $B$-hist exploits TRank \cite{6} to recognize named entities and to assign a unique entity type to each of them (e.g., Tom Cruise $\rightarrow$ American Actor). Candidate entity types are those used by DBpedia, Freebase, and YAGO. Thanks to this, we can define the distance $\delta$ between two Web pages as

$$\delta(p_0, p_1) = \frac{\left(\sum_{(t,t')\in\tau(p_0)\times\tau(p_1)} dist(t, t')\right)}{|\tau(p_0) \times \tau(p_1)|} \quad (1)$$

where $dist(t, t')$ is the distance between two entity types $t$ and $t'$ in the TRank type hierarchy, defined as the sum of the number of steps in the hierarchy needed to reach their least common ancestor starting from each one of them. We finally use $\delta$ to cluster pages by using a variant of the $k$-means clustering algorithm in which the centroid of each cluster is a list composed by the $n$ most frequent types identifying its sessions. The main property of the variation of $k$-means we use is that there is no need to specify the number $k$ of clusters. Rather, we specify a threshold $\Delta$ and, each time a page we want to cluster is further than $\Delta$ from every existing cluster, a new cluster is created. With this approach, we regroup pages on similar entities creating thematic clusters for the user.

We analyzed different values for $\Delta$ over one browsing history of 7 days from one user. The number of generated clusters is shown in Figure 3.

As it can be seen, the number of generated clusters steadily decreases until a plateau is reached at $\Delta = 10$. At that point almost all the Web pages are assigned to the same cluster. The user who provided Web history data for this experiment selected the clusters produced by setting $\Delta = 6$ as the best clustering result. Given this result, we choose to keep this setting for all of our further experiments.

3.3. B-hist Search Dashboard

After the process described above, the Web pages’ contents and generated metadata are available for search via the $B$-hist dashboard (see Figure 1). The users are first presented with a summary of their two latest weeks of browsing activities. Each element on the dashboard serves both for filtering and for providing information as the information displayed in each component is updated dynamically after each click.

User interactions are handled by four different components:

- A search box powered by entity suggestion
- A time-based focus with interval selection
- An entity-centric filter
- Groups of semantic sessions.

The main entry point to search through one’s personal browsing history is the familiar search box. The $B$-hist search box is powered by a query auto-completion feature that suggests entities appearing in the user’s browsing history based on the query he/she is typing in the box. Such a functionality can be exploited by users as a way to self-select the sessions they are most interested in. Thus, user-initiated session clustering becomes an alternative to the algorithmic clustering that $B$-hist precomputes and proposes on its middle panel.

A second possibility to filter results is based on the time dimension (left panel): the default view focuses on the previous two weeks, though the user can change it by selecting a different interval in the calendar (with a minimum granularity of one day).

The third option to filter results is to select an entity or an entity type in the left panel (below the calendar). Thanks to this panel, the user can specify which entity (or entity type) he/she is interested in and see the clusters, time periods, and URLs most relevant to it.

The forth option to interact with the user history is the session clusters in the middle panel: first, the user is presented with a set of clusters that are relevant to the current filters. Then, if the user clicks on a cluster, he/she will be presented with the set of entities belonging to the pages in that cluster.
The right panel of the dashboard contains a list of URLs ordered by access time, which reflects the currently selected filters.

Each update to the filters will automatically update the results in the other components of our user interface. We expect the user interaction with B-hist to finish either when the intended URL has been found and clicked (i.e., re-finding activity) or simply when the user identifies an entity or entity type he/she was trying to recall using B-hist.

3.4. B-hist Query Processing

Each interaction of the user with the B-hist’s dashboard is managed by making a POST request to the back-end. The request contains the current constraints specified by the user by interacting with the dashboard and the back-end returns an object specifying the new content of each component of the UI. Together with this the back-end returns the list of words that are auto-completed while typing in the search box, that is, all the names and types of the entities satisfying the current constraints and the names of the shown clusters, if any. All the data is exchanged in JSON format.

4. Experimental Validation

In this section, we present the result of an online survey conducted to support design choices. Specifically, we asked more than 200 Web users which functionalities they would appreciate in a tool like B-hist. We also present an experimental comparison of different clustering techniques for Web browsing sessions.

4.1. On-line Survey about Web History Search

In order to validate our design choices, we run an online survey involving more than 200 Web browser users. In terms of demographics, our population includes 74 female and 175 male users with average age of 29.8. The geographical distribution of the population is dominated by India with 144 users and the USA with 67 users.

In the survey, after some basic demographic questions, we asked users about their Web browsing experience (i.e., how much time they spend browsing the Web) and about their Web history search experience (i.e., how frequently they search in their history). Finally, we asked to rate in a scale from 1 to 5 (where 1 means “useless” and 5 is “very useful”) different new functionalities that could improve their Web history search experience. Moreover, we let the user provide other desired functionalities as free text.

The majority of users declared to search in their browsing history more than once a day and to browse the Web more than 3h per day but less than 8h per day. The different pieces of information we tracked in our survey are listed in Table 1.

Figure 4, 5, 6, and 7 summarized the perceived utility of the various features to improve Web history search.

We observe that the most interesting features according to the users are Clusters and Sessions. These are two different approaches for grouping visited Web pages either on the topical dimension or based on coherent user activities. It is evident that users need to be supported in some way when

---

4We recruited them on Amazon MTurk.
Searching over their browsing history as they access past information (also known as re-finding [18, 19]). These findings motivate our UI design choices where in the central panel we display clustered browsing sessions that are a possible starting point of the user interaction.

Contrary to our hypothesis, entity-centric information access was not considered useful by as many users. Figure 8 shows the breakout of perceived utility of entities based on the user history search activity. We can clearly observe that the more often users use browsing history search functionalities, the more they perceive as useful the ability to find information based on which entities are present in the browsed Web pages. Moreover, entities are a key component of the clustering algorithms used by B-hist (see Section 3.2).

In conclusion, the results of the survey we conducted supports our design choices to provide functionalities that are missing in current Web browsing history search as available in standard browsers.

4.2. Evaluation of Browsing Session Clustering

In the following, we compare different clustering strategies based on well-known techniques against the algorithm described in Section 3. In order to perform this evaluation, we built a tool that extracts the user browsing history, runs different clustering approaches, and shows the results to the user, who then annotates the produced output hence creating relevance judgments for the clustering. In this way, we are able to ask users to evaluate different solutions without asking them to disclose their browsing history to anyone.

In particular, the methods we compare are:

**Tf-idf clustering**, that is, the application of the $k$-means clustering algorithm to the term vectors representing each Web-page contained in the Web browsing history. In that context, each term vector represents a Web-page while each of its components is the tf-idf weighting of a word appearing in the page. Stop-words are removed and the remaining tokens are stemmed by the Loving stemming algorithm.

**LDA-based clustering**, in which we use Latent Dirichlet Allocation (LDA) [20, 21] to create the clusters. In particular, we choose as many topics as clusters and assign each document to the cluster corresponding to the dominant topic in the page.

**B-hist clustering**, the entity-centric clustering approach proposed in Section 3.
We note that, while tf-idf and LDA clusterings require a fixed number of clusters (the $k$ used in $k$-means and the number of topics, respectively), our B-hist clustering does not require such an input. In order for our evaluation to be fair, we first run B-hist clustering and then use the number of clusters it produces both as $k$ in tf-idf as well as the number of topics in LDA.

We applied all the considered clustering algorithms over the last four days of history for 5 different individuals and asked them to judge the quality of the generated clusters on a scale from 1 to 5 (where 1 means “very bad” and 5 corresponds to “very good”). The experiment was run on the user-machine running a script that takes the browsing history, runs all the clustering algorithms, and displays the judging interface. Users then could judge the quality of the clusters without knowing which cluster was generated by which algorithm. Clusters were displayed as a set of URLs with their HTML Title. Once finished, users sent back a generated file with cluster id and quality judgment pairs. In that way, the browsing history remained private to the users while providing us with an empirical assessment of the quality of the clusters allowing us to compare the different clustering strategies.

Table 2 shows the average ratings given by the users for the various approaches.

From the previous experiment, we observe that the most appropriate clustering algorithm is based on tf-idf. This shows the fact that page content is a good indication of web page clusters and that users find it good enough for re-finding information that they have accessed in the past. This result may be also explained by the fact that we used the most recent browsing history. Since human memory fades out as time goes on, the same experiment on an older set of web pages may favour a different clustering algorithm.

5. Conclusions

In this paper, we described $B$-hist: The first system offering semantic functionalities over Web history search. The main contribution of this work is the enrichment of classic Web history search by means of entity-centric search, entity type and time range filters, and clustering based on web browsing sessions. We have shown how modern technologies that have been successfully applied to Web search can be ported to enrich the web browsing history search experience. Our results show that users may appreciate such functionalities and could improve user experience.

The current version of $B$-hist runs on the user machine: In order to preserve the user privacy, no data is ever sent to any third-party. However, we envision a server-side version of the system using scalable storage, indexing and processing techniques (e.g., Apache Solr and Hadoop as described in [6]). In such a setting, users would be sharing their browsing activities (as they already do by using any of the commercial Web browsers) and would obtain additional functionalities. For example, one could provide personal analytics functionalities (e.g., ‘How do I spend my time online?’) and recommendations using, for instance, collaborative filtering approaches that correlate data across similar $B$-hist users.

We also envision a ‘forget’ functionality as not all information accessed online stays relevant on the
Table 2. Average rating (where 1 means “very bad” and 5 corresponds to “very good”) given by users to clusters generated by different algorithms. Users are denoted by A, B, C, D, E.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>tf-idf</td>
<td>4.78</td>
<td>4.88</td>
<td>2.18</td>
<td>2.55</td>
<td>4.80</td>
<td>3.84</td>
</tr>
<tr>
<td>LDA</td>
<td>3.94</td>
<td>3.26</td>
<td>2.83</td>
<td>3.27</td>
<td>2.70</td>
<td>3.20</td>
</tr>
<tr>
<td>Entity-centric</td>
<td>4.17</td>
<td>4.33</td>
<td>1.38</td>
<td>2.78</td>
<td>3.95</td>
<td>3.22</td>
</tr>
</tbody>
</table>

long term. By analyzing user interaction with B-hist, the system would learn which type of information the user is most interested in and would consider other types of information as less important (similarly to the way human memory works).

Finally, further investigation are necessary to validate our UI design choices. We plan to run a large scale user study with the users who installed and used the system.

6. Acknowledgements

This work was supported by the Swiss National Science Foundation under grant number PP00P2_128459, and by the Häslerstiftung in the context of the Smart World 11005 (Memor1es) project. We also thank for their help and feedback Martin Grund, Eugenia Martin, Ruslan Mavlyutov, and Vincent Pasquier.

References


