Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science

Web Semantics: Science, Services and Agents on the World Wide Web 00 (2014) 1–9

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

Michele Catasta[a,*], Alberto Tonon[b], Gianluca Demartini[b], Jean-Eudes Ranvier[a], Karl Aberer[a], Philippe Cudré-Mauroux[b]

[a] *EPFL—Switzerland*
[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 effectively 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 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.

Keywords: browsing history, personal memories

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 is powered by semantic data: 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. This is possible thanks to structured knowledge bases and LOD datasets such as Freebase 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. Most browsers provide a very limited keyword-based search over previously visited pages, which has not changed much in the last 20 years.

In this paper, we present a new system that lets users search over their personal Web browsing history in a more effective way. The goal of our system, called *B-hist* (standing for *'Better history'*), is to bring entity-centric access to personal browsing

*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)

activities thanks to semantic technologies such as the ones we developed in our recent pieces of work [1, 2] for entity disambiguation and entity type selection.

By mining entities in Web pages and leveraging their types to cluster pages in meaningful groups, we allow the user to access his/her Web history from multiple entry points: Users can type queries which get autocompleted 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

In [3, 4] 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 [5] Mayer and Bederson present a system that allows the user to organize his/her 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 UI we use a more user-friendly representation consisting of pictures of the main entities appearing in the session.

In [6] 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 [7] 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.

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.

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 backend architecture.

3.1. System Components

3.1.1. Chrome Browser Extension

The initial data collection is handled by a Web browser extension[3], 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

[1] `https://wiki.mozilla.org/Pancake`
[2] `http://cottontracks.com/`
[3] At this point, we provide an extension for the Chrome browser.

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 user can start looking for information in his/her browsing history (see Figure 1).

3.1.2. B-hist *Data Processing*

Once the raw HTML data is gathered from an accessed page, it goes trough our TRank [2] processing pipeline (see Figure 2) where the main textual content is kept (using approaches from [8]), entities are extracted (using a Conditional Random Field approach trained on a news corpus [9]), and entity types are selected (using approaches from [2]). Such metadata on the Web pages is stored and indexed in *B-hist* (see Figure 2).

To store and index data (e.g., timestamps and cluster information) we use both an inverted index (i.e., Apache Lucene[4]) and a lightweight DBMS (i.e., SQLite). 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 term.

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 grouping pages in coherent sessions with a common user intent.

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 [2] to recognize named entities and to assign a unique entity type to each of them (e.g., Tom Cruise → American Actor). Candidate entity types are those used by DBpedia, Freebase, and YAGO. Thanks to this, we can define the distance δ 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, and is 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 δ 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 Δ and, each time a page we want to cluster is further than Δ from every existing cluster, a new cluster is created. With such approach we group together pages about similar entities creating thematic clusters for the user to browse.

We analysed different values for Δ over one browsing history of 7 days from one user. The number of generated clusters is shown in Figure 3.

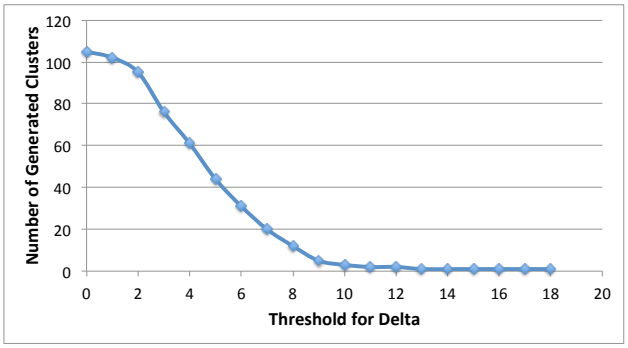

Figure 3. Number of clusters generated with different values of the Δ threshold.

Given the previous result, we select as value for *Delta* 6 which produces a reasonble number of clusters for *B-hist* users.

3.1.3. 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 user is first presented with a summary of the latest two weeks of browsing activities. Each element of 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

[4] http://lucene.apache.org/

Figure 1. Welcome screen of the *B-hist* dashboard. The user has access to his/her browsing activities in the previous two weeks aggregated over time, entities, and sessions.

- 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 autocompletion 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 used 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 is on the previous two weeks but 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 which 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*.

On-line availability. The system is accessible online at `http://mem0r1es.io` where we provide a screencast demonstrating the end-user *B-hist* dashboard. Moreover, for the purpose of judging our system at the Semantic Web Challenge, we provide access to an online deploy of the *B-hist* dashboard which allows to search over a fictitious browsing history. GD: decide if we want this

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.

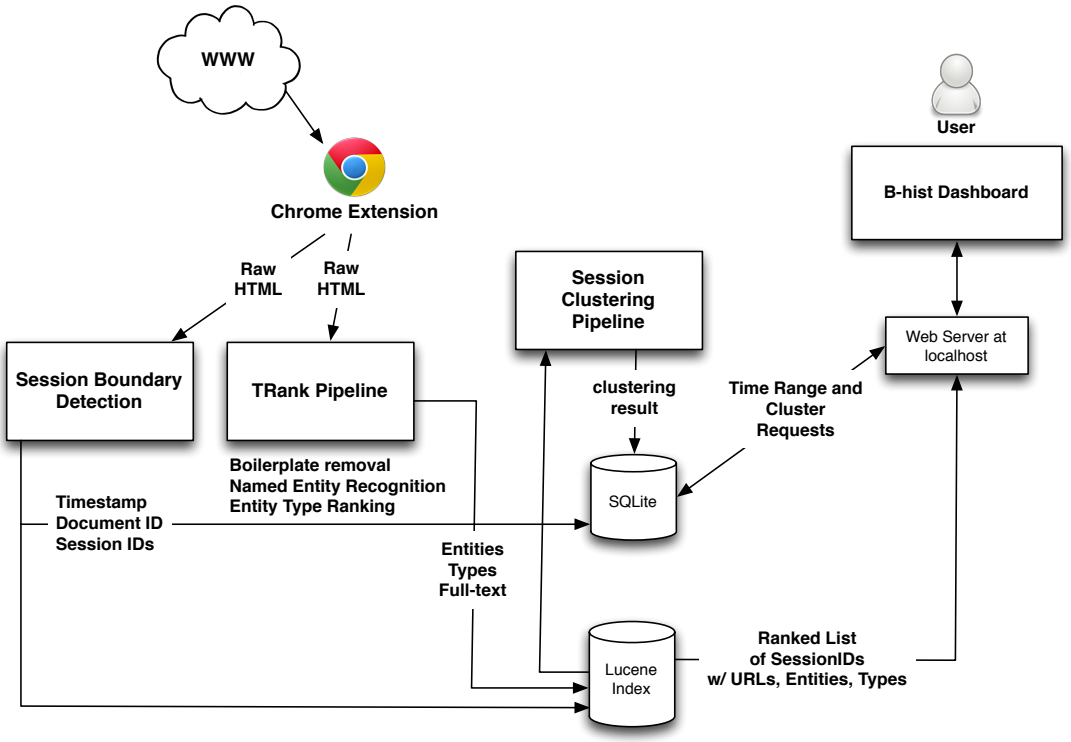

Figure 2. *B-hist* data processing architecture: First the raw HTML of a Web page visited by the user is provided by the browser plugin. Next, the page is processed through boilerplate removal, named-entity recognition, entity type selection, and clustered in an existing or new session. Then, the generated metadata is stored and indexed. Finally, sessions are clustered together in semantically coherent groups. The user's access to information happens via the *B-hist* dashboard.

4.1. On-line Survey about Web History Search

In order to validate our design choices, we run an on-line survey involving more than 200 Web browser users[5]. In terms of demographics, our population includes 74 female and 175 male users with average age of 29.8. The geographical distrubution of the population includes India with 144 users and USA with 67 users as most represented countries.

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 means very useful) different new functionalities that could improve their Web history search experience. Moreover, we let the user provide other desidered functionalities as free text.

[5]We recruited them on Amazon MTurk.

The majority of users declare 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 features we asked about in our survey are listed in Table 1.

Figure 4, 5, 6, and 7 show the perceived utility of different functionalities to improve Web history search.

We can observe that the most interesting features for 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 searching over their browsing history as they access past information (also known as re-finding [10, 11]) remembering the topic or the activity they were carring on.

These findings motivate our UI design choices where in the central panel we display clustered browsing sessions which are a possible starting point of the user interaction.

Sessions: group Web pages by related activities (e.g., buying a digital camera)
Entities: display the persons, locations, and organizations mentioned in the pages you have visited (e.g., Tom Cruise)
Calendar: display the browsing activity intensity over time (e.g., more on Sunday than on Friday)
Clusters: group of Web pages by topic (e.g., all pages about football)

Table 1. List of functionalities included in the on-line survey.

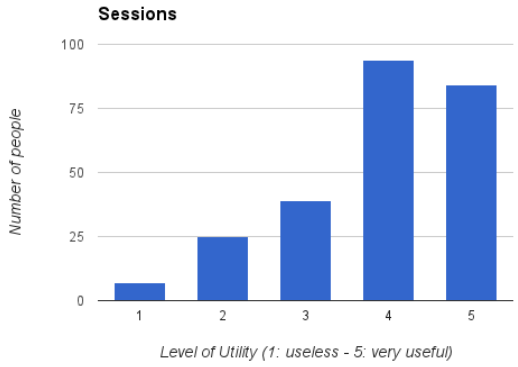

Figure 4. Utility of Sessions in Web History Search

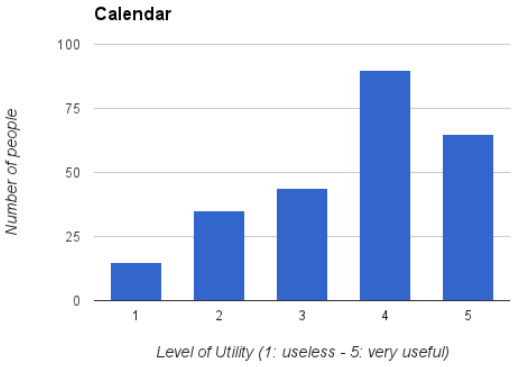

Figure 6. Utility of Calendar in Web History Search

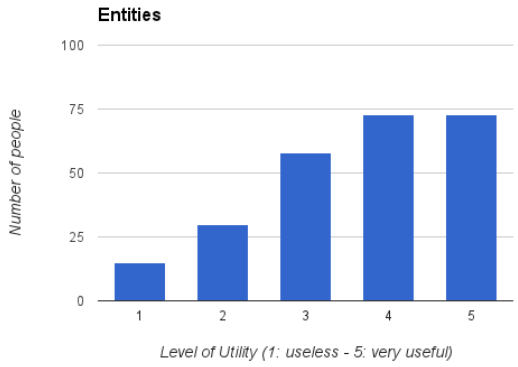

Figure 5. Utility of Entities in Web History Search

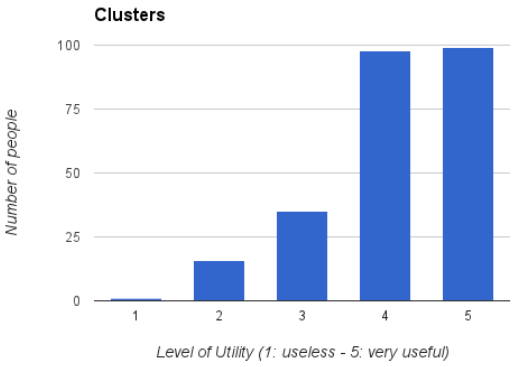

Figure 7. Utility of Clusters in Web History Search

Contrary to our hypotesis, entity-centric information access of browsing history was not consideredy 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 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 3.1.2).

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

4.2. Evaluation of Browsing Session Clustering

Here we compare different clustering strategies based on well-known methods against the algorithm described in Section 3. In order to perform such evaluation we built a tool that extracts the user browsing history, runs different clustering approaches and shows the results to the user, who annotates the produced output thus creating relevance judgments for the clustring output. In this way we are able to ask users to evaluate different solutions without the need for 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. Each term vector represents a Web-page and 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, that is, we use Latent Dirichlet Allocation (LDA) [12, 13] to create the clusters. In particular, we have as many topics as clusters and we assign each document to the cluster corresponding to the dominant topic in the page.

B-hist clustering, the entity-centirc clustering approach proposed in Section 3.

Notice 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), B-hist clustering does not require such input. In order for our evaluation to be fair, we first run B-hist clustering and then we use the number of clusters it produces both as k in tf-idf as well as the number of topics in LDA.

We applied the compared clustering algorithms over the last four days of browsing history for 5 different people and asked them to judge the quality of the generated clusters in a scale from 1 to 5 (where 1 means vary bad and 5 means very good). The experiments was run on the user machine running a script that takes the browsing history, runs all the clustering algorithms, and produces the juding interface. Users then judge the cluster quality 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 this way the browsing history remains private to the user who provides us with an assessment of clustering quality that allows us to compare the different clustering strategies.

Table 2 shows the average rating given by the different users to the clusters generated by different approaches.

From the previous experiment we have observed that the most appropriate clustering algorithm is tf-idf and we are thus using as component of *B-hist*.

5. Conclusions

In this document, we described *B-hist*: The first semantically-enriched Web browsing activity search end re-finding tool.

The current version of *B-hist* runs on the user machine: In order to preserve his/her 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 [2]). 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

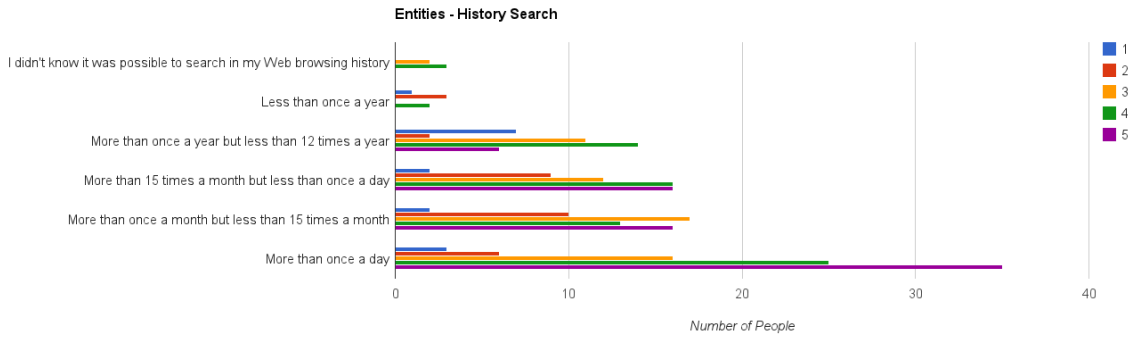

Figure 8. Utility of Entities based on Search activity. Utility is expressed on a scale from 1 to 5 where 1 means useless and 5 means very useful.

Algorithm	A	B	C	D	E	Avg
tf-idf	4.78	4.88	2.18	2.55	4.80	3.84
LDA	3.94	3.26	2.83	3.27	2.70	3.20
Entity-centric	4.17	4.33	1.38	2.78	3.95	3.22

Table 2. Average rating given by users to clusters generated by different algorithms (users are denoted by A, B, C, D, E).AT: Add Michele's eval?

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 (i.e., similarly to the way in which the human memory works).

6. Acknowledgements

This work was supported by the Swiss National Science Foundation under grant number PP00P2_128459, and by the Haslerstiftung in the context of the Smart World 11005 (Mem0r1es) project. We also thank for their help and feedback Martin Grund, Eugenia Martin, Ruslan Mavlyutov, and Vincent Pasquier.

References

[1] A. Tonon, G. Demartini, P. Cudré-Mauroux, Combining inverted indices and structured search for ad-hoc object retrieval, in: SIGIR, 2012, pp. 125–134.

[2] A. Tonon, M. Catasta, G. Demartini, P. Cudré-Mauroux, K. Aberer, TRank: Ranking Entity Types Using the Web of Data, in: International Semantic Web Conference, 2013.

[3] A. COCKBURN, B. MCKENZIE, What do web users do? an empirical analysis of web use, International Journal of Human-Computer Studies 54 (6) (2001) 903–922. doi:10.1006/ijhc.2001.0459.
URL http://www.sciencedirect.com/science/article/pii/S1071581901904598

[4] B. McKenzie, A. Cockburn, An empirical analysis of web page revisitation, in: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 5 - Volume 5, HICSS '01, IEEE Computer Society, Washington, DC, USA, 2001, pp. 5019–.
URL http://dl.acm.org/citation.cfm?id=820757.821892

[5] M. Mayer, B. B. Bederson, Browsing icons: A task-based approach for a visual web history, Technical Report.
URL http://drum.lib.umd.edu//handle/1903/1167

[6] Y. Shirai, Y. Yamamoto, K. Nakakoji, A history-centric approach for enhancing web browsing experiences, in: CHI '06 Extended Abstracts on Human Factors in Computing Systems, CHI EA '06, ACM, New York, NY, USA, 2006, p. 1319âĂŞ1324. doi:10.1145/1125451.1125696.
URL http://doi.acm.org/10.1145/1125451.1125696

[7] H. Obendorf, H. Weinreich, E. Herder, M. Mayer, Web page revisitation revisited: Implications of a long-term click-stream study of browser usage, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07, ACM, New York, NY, USA, 2007, pp. 597–606. doi:10.1145/1240624.1240719.
URL http://doi.acm.org/10.1145/1240624.1240719

[8] C. Kohlschütter, P. Fankhauser, W. Nejdl, Boilerplate detection using shallow text features, in: Proceedings of the third ACM international conference on Web search

and data mining, WSDM '10, ACM, New York, NY, USA, 2010, pp. 441–450. doi:10.1145/1718487.1718542.
URL http://doi.acm.org/10.1145/1718487.1718542

[9] J. R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 363–370. doi:10.3115/1219840.1219885.
URL http://dx.doi.org/10.3115/1219840.1219885

[10] H. Bruce, W. Jones, S. Dumais, Keeping and re-finding information on the web: What do people do and what do they need?, Proceedings of the American Society for Information Science and Technology 41 (1) (2004) 129–137.

[11] S. K. Tyler, J. Teevan, Large scale query log analysis of re-finding, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10, ACM, New York, NY, USA, 2010, pp. 191–200. doi:10.1145/1718487.1718512.
URL http://doi.acm.org/10.1145/1718487.1718512

[12] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
URL http://dl.acm.org/citation.cfm?id=944919.944937

[13] D. Newman, A. Asuncion, P. Smyth, M. Welling, Distributed algorithms for topic models, J. Mach. Learn. Res. 10 (2009) 1801–1828.
URL http://dl.acm.org/citation.cfm?id=1577069.1755845