STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXII**, Number 2, 2017
DOI: 10.24193/subbi.2017.2.05

ANALYZING THE USEFULNESS OF THE USER'S BROWSER HISTORY FOR GENERATING QUERY SUGGESTIONS

IOAN BĂDĂRÎNZĂ

Abstract. A very useful feature of search engines that helps users while they browse the internet, where, they very often, try to satisfy an information need, is *query suggestion*. This mechanism shows the user a list of possible queries from where he can choose and be able to perform a search easier and faster. In this paper we tried to assess the usefulness of a user's recent web browsing history for generating new query suggestions. We performed a one month experiment in which we collected browsing history logs of several users and searched query terms submitted by those users to Google (using a Chrome plugin) and found that approximately 32% of the queries submitted can be predicted from the user's browsing history.

1. Introduction

Searching for information on the web can be very difficult sometimes. There are a lot of users that do not know what terms to enter in a search input of a search system to better describe their information need. In [8, 15] we can see that most of the search queries are very short, one or two words on average and in [9, 16] we can see that these words are ambiguous. In order to help the user when performing a search, most search engines like Google, Yahoo!, Bing and others, provide query auto-completion and query suggestions. In order to better explain how search suggestions are generated, we will first try to describe how query auto-completion works. In almost all modern browsers, search engines and text editors we can see how, after we start typing words, it automatically tries to predict what we actually want to type. These are called 'predictive auto-completion systems' where the candidates are matched against the prefix using information retrieval techniques and also Natural Language Processing techniques. This auto-completion is actually

Received by the editors: November 2, 2017.

2010 *Mathematics Subject Classification.* 68U10, 94A08.

1998 *CR Categories and Descriptors.* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

Key words and phrases. query suggestion, browsing history, personalized suggestions.

the highest ranked suggestion from a suggestions list. The query suggestions list is a list that contains from eight to ten words (or group of words), which are usually prefixed with the subquery that the user is typing, items that are extracted from a huge log of queries submitted by all users. A very well known technique of extracting suggestions from a common query log is called Most Popular Completion, which we'll describe more in the next section of this paper.

The main focus of this paper is to analyse how user's personal browsing history and submitted query history are impacting the query suggestion list. In order to do this, we first created a Chrome extension which collects information about what web pages is the user visiting, what queries he submitted to Google and what suggestions Google returned for his subquery. By subquery we refer to the prefix of the query he started to type in the search input. Using this history, we perform an analysis on how important is this history on ranking future query suggestions. Moreover, we can later create user profiles that would improve the query suggestions offered by a search engine, like Google.

2. Related work

Query auto-completion. Auto-completion is used almost in all information retrieval engines. We have all seen how, in the search boxes of search engines, after we start typing the first character of our query, we immediately receive a possible auto-completion which will save us keystrokes when trying to fulfil an information need. What stands at the base of all these auto-completions is mostly the query logs of those particular search engines individually. We can see this kind of research in [2], [7], [5], [11], [10]. These approaches, do return pretty good suggestion lists but they lack a very particular thing, which is 'context'. This 'context' is composed by the immediately preceding queries that a user submitted. In [3], Bar-Yossef and Kraus demonstrated how recent user queries can significantly improve query auto-completion. They compare their results with the *Most Popular Completion (MPC)* which is one of the popular techniques for query suggestion. In [3], they say that the basic principle of MPC is users wisdom. This means that, if a particular query was used by a lot of users in the past, it is more likely that, that particular query will be the first candidate as an auto-completion. We can take, for example, a very popular and well known at the moment this article was written, social media website, "facebook". If we are trying to start typing letter "f" on www.google.com, the first auto-completion that we can see is "facebook" and that's because a lot of people are performing this particular query on google.com (see Fig. 1). In short terms, MPC is actually ranking suggestions based on their popularity. Let's say that we have a search log with

FIGURE 1. Typing letter "f" in Google's search input with all its suggestions.

previous queries $QLog$, a subquery (or the prefix of the intended query) sq and a list of query-completions $QC(sq)$, where all the items are starting with the desired subquery. Using MPC formula [23], we can calculate a rank for all items in $QC(sq)$ and order these items by their rank:

$$MPC(sq) = argmax_{q \in QC(sq)} w(q),$$

$$w(q) = \frac{freq(q)}{\sum_{i \in QLog} freq(i)}$$

where $freq(q)$ is actually the number of ocurences of query q in $QLog$.

This formula is a *Maximum Likelihood Estimator*, which in [3], Bar-Yossef and Kraus, improve this popularity based algorithm and also take into consideration the previous queries of the user which are considered as 'query context'. They named this approach *NearCompletion*, where they compute similarity for this context and improve MPC and demonstrate using Mean-Reciprocal-Rank method, that the context of a query is very important when trying to generate suggestions. However, in their papers they only consider the query history of the user and not the personal browsing history of the user which is what we analyse in this paper.

Query Suggestion. Query suggestion and query auto-completion are very similar. The main scope of both of them is to save user keystrokes when performing a search. Query suggestion is an enhanced, proposed query that the user might be looking for, whereas an auto-completion is the possible query term that the user might want to type immediately after he started typing the first letter. Usually, auto-complete happens in the same search input where the user is writing his query and has to press either "enter key" or "right arrow key" to accept it; whereas auto-suggestion, usually appears as a list in the

form of a drop-down from where the user can either press the "down-arrow-key" or perform a mouse click to select it the desired suggestion. Both of these approaches are a real boost to the usability of search engines. Basically, we can say that auto-completion is the first item from the query suggestions list. In [4], they proposed a context aware query suggestion approach by mining click-through data and session data. First, they group similar queries into concepts and represent them on a bipartite graph. After this offline step, in an online step they will take the user query and find the concept for it in the graph and return the queries from that concept as suggestions. Another paper where click through data was analysed and used for ordering the suggestions is [14], where they demonstrate that the higher a suggestion is present in a suggestions list, tends to attract more click. In [6] Jiang et al. are reformulating the query by analysing how users previously reformulate their queries then adding words in the query and define a set of features which were applied using the LambdaMart [12] learning to rank algorithm. Others [13] have tried to apply probabilistic models, like Markov Process to predict what user's query will be immediately after he starts typing.

Personalized search. All the above papers do not consider the recent browsing history of the user when offering query suggestions to the user. Our main focus of this paper is to analyse the usefulness of the user's recent browsing history for query suggestions which will allow us to create a personal profile for each user and use that profile when ranking query suggestions. Personalized search, in general attracted attention of a lot of researchers, [18, 19, 20, 21, 22]. Each and every study showed that user's personal query history is very important in search systems. Let's take for example the very well known query "ajax". This query has three meanings that we are aware of: one would be the Dutch football team "Ajax", another one would be the cleaning product "Ajax" and the last one would be "Asynchronous JavaScript and XML" used in web development. In [1, 24, 25] we can see that these kind of queries are used by users pretty often. If we do not know anything about user's previous searches and interests, we could not know which result represents user's information need. In general, the way personalized search applies in auto-completion and query suggestion is by saving each query that a user used at a particular point in time, then use all this history in ranking query suggestions. In [17], we can see how Bennett et al. demonstrated that the long term query history is very useful when the users starts his search session and the short term query history is more relevant when the search session evolves. Matthijs and Radlinski, in [18] used a browser plugin to collect browsing history and used that history to re-rank search results and demonstrated that the returned results are more relevant to the user. Others, like Shokouhi in [23], went even

further with personalisation and divided users into categories based on their age, gender and region and demonstrated that all these features have an impact on the suggestion that a user is waiting for when trying to search. For example, after typing letter 'i' in a search input, the most selected suggestion by male users is 'imdb' whilst female users were choosing 'indeed.com'. Another interesting example, from [23], is that users below 20 years mostly selected 'full house' after typing letter 'f' whilst the users above 50, selected 'florida lottery'.

All the above papers either consider the global or personal query history (measured at the search engine) or they use a form of browsing history, but for re-ranking search results returned by the search engine [18]. In contrast, we consider the personal browsing history of the user in order to provide better query suggestions. In this paper, we will present an experiment that validates the hypothesis that the user's recent browsing history is important for new query suggestions and a significant number of new queries can be predicted from the user's recent browsing history.

3. Architecture of the browser extension

In order to collect browsing history and submitted queries to Google search we have built a Chrome extension, named ***User History Collector***. The reason for collecting only Google searches is the fact that according to ***comScore*** in [26], in February 2016, out of the total explicit core searches performed on web, 64% were Google searches. The other part of 36% is divided between Bing, Yahoo, Ask Network etc. We choose to build a Chrome extension, and not an extension for other browsers, because according to latest *Browser Statistics* [27] from October 2017, made by *www.w3schools.com*, 76.1% of users are using a Chrome browser.

The entire extension is written in *javascript* which makes REST calls to some APIs that are persisting all user information in a MySQL database for later offline analysis. In Fig. 2 we can see the components of the extension and how data flows from one component to another. The *background script* and *content script* are actually components of a Chrome extension. The *content script* is a way of the extension to interact with webpages that are opened in a tab; it can be viewed as a part of the webpage, which is executed after the page is loaded. We use this component to extract the content of webpages. The *background script* is a component that holds the logic of the extension. We use this component to parse the data and send it to a server by making REST calls. The way our extension functions is, whenever a new page is loaded, the *content script* will be executed and based on the webpage, it will do the

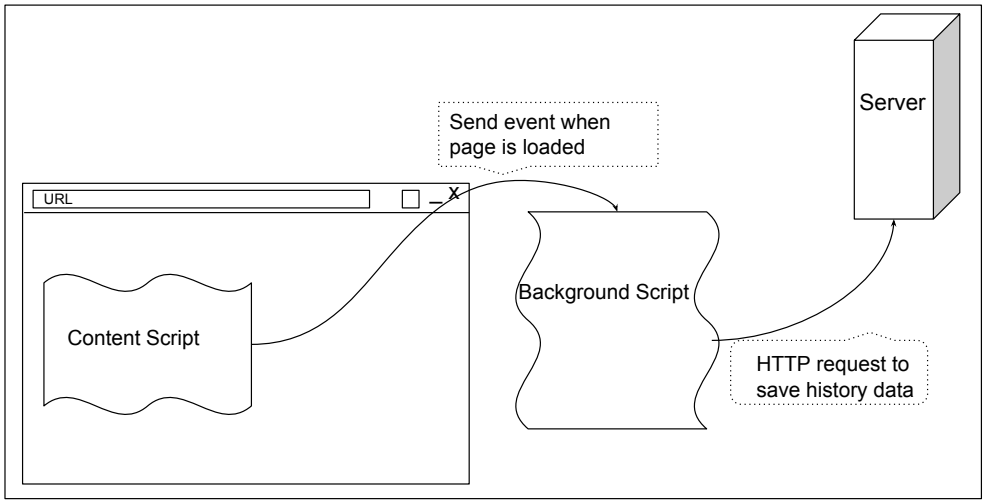

FIGURE 2. User History Collector components diagram

following (all webpages that are email pages, facebook pages and other pages that may contain personal information will not be analysed, will be ignored):

(1) If the URL of the page does not start with *"www.google."*, it will interpret it as a new webpage that was viewed and will extract the actual text from the HTML document and, alongside with page URL and page title, it will pass it to the *background script*. The *background script* will split the text in terms, will eliminate stop words and will calculate the term frequency for each unique word. After this step, it will make an Ajax HTTP request to a server which will store all the history data for later analysis.
(2) If the URL of the page, does match *"www.google."*, it means the user trying to perform a Google search. In this case:
    (a) For each key pressed in Google's search input, the *content script* will extract the value of the search input and the list of suggestions provided by Google for the written subquery. This information is passed to the *background script* which will send it to the server.
    (b) When user finishes to type the desired query and submits it to Google, that particular query is passed to *background script* which will send it to the server.

For all information that is passed to the *background script*, this will associate a unique identifier (which is generated once when the user installs the extension) before making the request to the server for persisting it.

4. Analysing collected data

Time period	Total number of clients	Total number of visited pages	Total number of Google queries
1 Month	14	14571	1847

Figure 3. Collected data

After having the extension running and collecting data for over a month, in table from Fig. 3 we can see that it gathered 14571 visited pages and 1847 queries that were submitted to Google from a number of 14 unique users that had the extension installed on their Chrome browser.

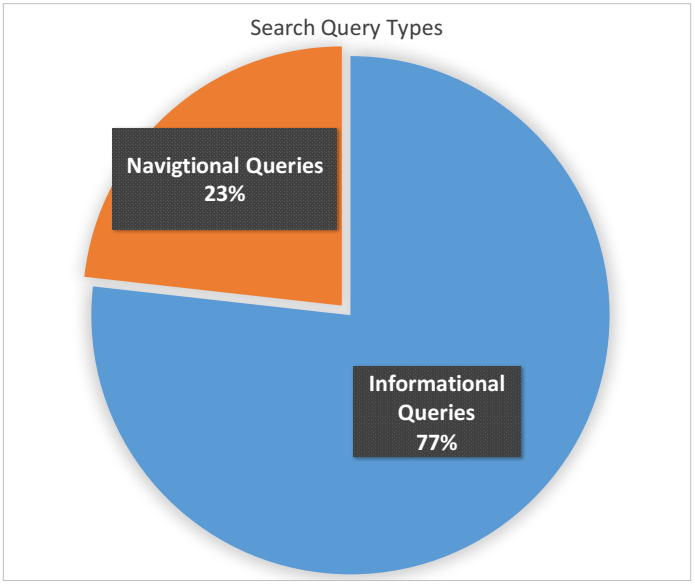

Figure 4. Query types

It is commonly accepted that search queries can be divided into two main types: *navigational queries* and *informational queries*. A *navigational query* is a search query entered by the user with the intent of finding a very particular

webpage. For example, a user might type "facebook" into Google's search input in order to find and navigate to "Facebook" website. Another example would be if the user wants to go to "Yahoo Mail", he might search for "yahoo mail" on Google, instead of directly typing the full address in the address bar. We can say that whenever a user submits a query to Google, and the URL of the first page that he navigates to contains all the terms from the query, the query is a *navigational query*. An *informational query* is a search query that can cover a very large topic, for which, the search engine can return a very large number of relevant results. When a user submits such a query to Google, he is looking for some information and not a particular website. For example, if the user submits the query "einstein birthdate", he is clearly looking for some information without caring the website he gets this from. In Fig. 4, which is built from the data collected by our Chrome extension, we can see that 77% of the queries are *informational queries* and 23% are *navigational queries*. We considered a query to be a *navigational query* if the URL of the first page, that is visited by the user, contains all query terms; all other queries that do not follow this rule are considered as *information queries*.

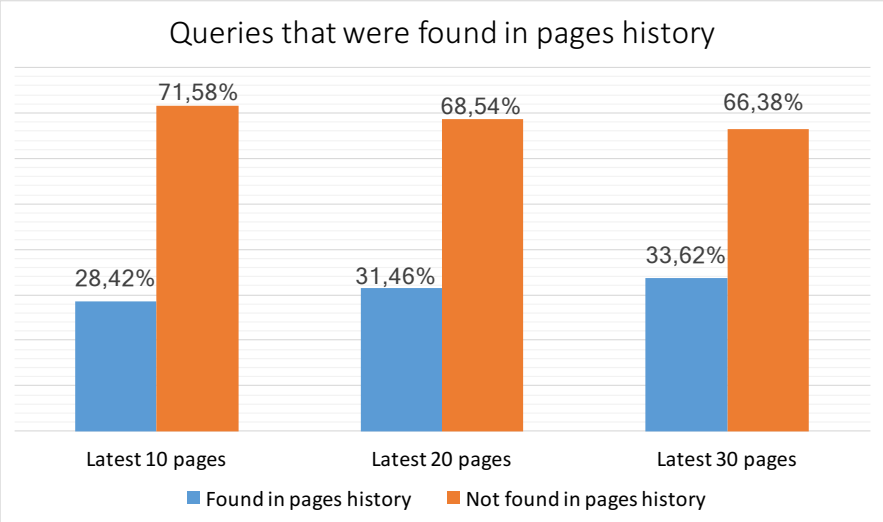

FIGURE 5. Relevance of browsing history

In Fig. 5 we analysed how many of the query terms of a query, can be found in webpages that were previously visited. If the query term appear in the URL of the page or in the title of the page, we no longer look within the actual content of the page because we have already found them. We made several

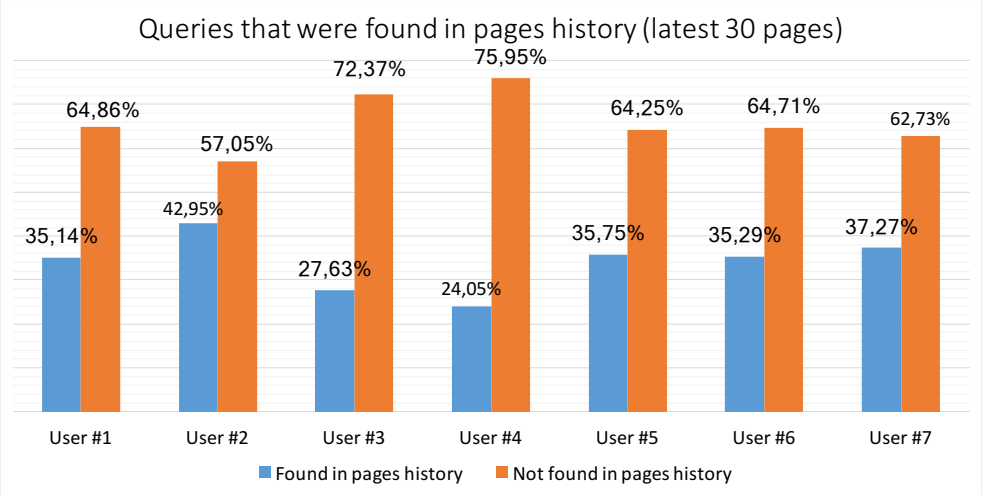

FIGURE 6. Relevance of browsing history (latest 30 pages) for particular users

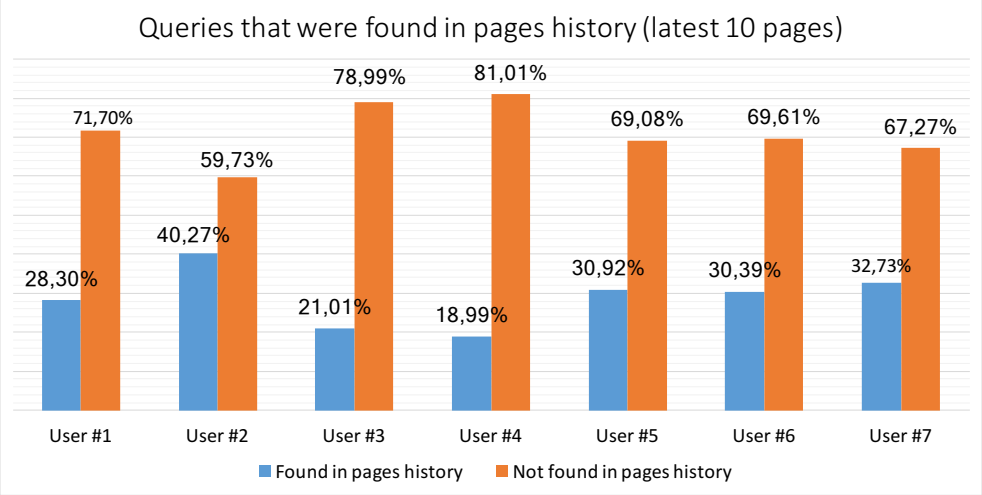

FIGURE 7. Relevance of browsing history (latest 10 pages) for particular users

tests related to the length of the recent history. First we considered the most recent history as latest 10 visited webpages. After this we increased this length to 20 pages, respectively 30 pages. We can observe how the number of the

queries, that have been found in the recent history, increases as the length of the history increases.

In Fig. 6 and Fig. 7 we divided these results, and display how many query terms, can be found in webpages that were previously visited for particular randomly selected users that have installed our extension. Calculating the 75th percentile for these values, we can say that 37.27% of the queries that a user submits to Google, are found in the recent 30 pages long history and 32.73% for a history containing only the latest 10 webpages visited.

5. Conclusions and future work

In this paper we have studied how recent browsing history of users can have an impact on the next queries that the user will submit to Google search. In order to do this, we created a Chrome extension, that collects data about all pages that a user visits, the queries that he submits to Google and also each subquery and the entire list of suggestions that Google returns for the subquery. After having the extension installed on users browsers and collecting data for a month, we analysed the data and concluded that, in lots of cases, this history can be used to extract suggestions and display them for the next time the user will want to submit a search query. A way to extract suggestions from previously visited pages would be to take the most recent and very short browsing history (most recent 2 - 3 pages which were visited in the last couple of minutes), calculate a weight for each word in the page and based on these weights and the prefix that the user will type next, extract the most representative words and offer them as personal suggestions.

References

[1] Ryen W. White and Steven M. Drucker. Investigating behavioral variability in web search. In Proceedings of the 16th International Conference on World Wide Web, WWW '07, pages 21-30, New York, NY, USA, 2007. ACM.

[2] Holger Bast and Ingmar Weber. Type less, find more: Fast autocompletion search with a succinct index. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, pages 364-371, New York, NY, USA, 2006. ACM.

[3] Ziv Bar-Yossef and Naama Kraus. Context- sensitive query auto-completion. In Proceedings of the 20th International Conference on World Wide Web, WWW '11, pages 107-116, New York, NY, USA, 2011. ACM..

[4] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pages 875-883, New York, NY, USA, 2008. ACM.

[5] Shengyue Ji, Guoliang Li, Chen Li, and Jianhua Feng. Efficient interactive fuzzy keyword search. In Proceedings of the 18th International Conference on World Wide Web, WWW '09, pages 371-380, New York, NY, USA, 2009. ACM.

[6] Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. Learning user reformulation behavior for query auto-completion. In Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, pages 445-454, New York, NY, USA, 2014. ACM.

[7] Mario Arias, Jose Manuel Cantera, Jesus Vegas, Pablo de la Fuente, Jorge Cabrero Alonso, Guido Garcia Bernardo, Cesar Llamas, and Alvaro Zubizarreta. Context-based personalization for mobile web search. In PersDB, pages 33-39, Auckland, New Zealand, 2008.

[8] Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Clustering user queries of a search engine. In Proceedings of the 10th International Conference on World Wide Web, WWW '01, pages 162-168, New York, NY, USA, 2001. ACM

[9] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In Proceedings of the 11th International Conference on World Wide Web, WWW '02, pages 325-332, New York, NY, USA, 2002. ACM

[10] Holger Bast, Debapriyo Majumdar, and Ingmar Weber. Efficient interactive query expansion with complete search. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07, pages 857-860, New York, NY, USA, 2007. ACM.

[11] Ryen W White and Gary Marchionini. Examining the effectiveness of real-time query expansion. Information Processing and Management, 43(3):685-704, 2007.

[12] Christopher J. C. Burges, Krysta M. Svore, Paul N. Bennett, Andrzej Pastusiak, and Qiang Wu. Learning to rank using an ensemble of lambda-gradient models. In Proceedings of the 2010 International Conference on Yahoo! Learning to Rank Challenge - Volume 14, YLRC'10, pages 25-35. JMLR.org, 2010.

[13] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Hongyuan Zha, and Ricardo Baeza-Yates. Analyzing user's sequential behavior in query auto-completion via markov processes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, pages 123-132, New York, NY, USA, 2015. ACM.

[14] Yanen Li, Anlei Dong, Hongning Wang, Hongbo Deng, Yi Chang, and ChengXiang Zhai. A two-dimensional click model for query auto-completion. In Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, pages 455-464, New York, NY, USA, 2014. ACM.

[15] Bernard J Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. Information processing and management, 36(2):207-227, 2000.

[16] Mark Sanderson. Ambiguous queries: Test collections need more sense. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 499-506, New York, NY, USA, 2008. ACM.

[17] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short and long-term behavior on search personalization. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, pages 185-194, New York, NY, USA, 2012. ACM.

[18] Nicolaas Matthijs and Filip Radlinski. Personalizing web search using long term browsing history. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, pages 25-34, New York, NY, USA, 2011. ACM.

[19] Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem, and Bilal Chebaro. A session based personalized search using an ontological user profile. In Proceedings of the 2009 ACM Symposium on Applied Computing, SAC '09, pages 1732-1736, New York, NY, USA, 2009. ACM.
[20] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In Proceedings of the 16th International Conference on World Wide Web, WWW '07, pages 581-590, New York, NY, USA, 2007. ACM.
[21] Ahu Sieg, Bamshad Mobasher, and Robin Burke. Web search personalization with ontological user profiles. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07, pages 525-534, New York, NY, USA, 2007. ACM.
[22] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, pages 449-456, New York, NY, USA, 2005. ACM.
[23] Milad Shokouhi. Learning to personalize query auto-completion. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, pages 103-112, New York, NY, USA, 2013. ACM.
[24] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Potential for personalization. ACM Trans. Comput.-Hum. Interact., 17(1):4:1-4:31, New York, NY, USA, 2010.
[25] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM '05, pages 824-831, New York, NY, USA, 2005. ACM.
[26] https://www.comscore.com/Insights/Rankings/comScore-Releases-February-2016-US-Desktop-Search-Engine-Rankings
[27] https://www.w3schools.com/browsers/

Faculty of Mathematics and Computer Science, Babeş–Bolyai University, Cluj-Napoca, Romania
E-mail address: ionutb@cs.ubbcluj.ro