A SURVEY OF COLLABORATIVE WEB SEARCH
Through Collaboration among Search Engine Users to More Relevant Results

Pavel Surynek
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, Prague, Czech Republic
pavel.surynek@mff.cuni.cz

Keywords: collaborative web search, social search, search engine, search results, collaborative filtering, recommender systems, system integration.

Abstract: A survey on collaborative aspects of web search is presented in this paper. Current state in full-text web search engines with regards on users collaboration is given. The position of the paper is that it is becoming increasingly important to learn from other users searches in a collaborative way in order to provide more relevant results and increase benefit from web search sessions. Recommender systems represent a rich source of concepts that could be employed to enable collaboration in web search. A discussion of techniques used in recommender systems is followed by a suggestion of integration web search with recommender systems. An initial experience with web search powering small academic site is reported finally.

1 INTRODUCTION AND MOTIVATION

Web search is an area of the information technology industry where artificial intelligence and particularly knowledge engineering techniques can be applied with potentially significant impacts. Currently users face a still increasing amount of data of many kinds that can be accessed through web (textual data, multimedia data, automatically collected data – CCTV records, etc...). Organizing and making these information accessible represents a continuous challenge despite the existence of powerful mainstream search engines like *Bing* (Microsoft Corp., 2013), *Google* (Google Inc., 2013), and *Yahoo!* (Yahoo! Inc., 2013), and also national ones like *Yandex* (Yandex Corp., 2013), *Baidu* (Baidu, Inc., 2013), *Naver* (NHN Corp., 2013), *Seznam* (Seznam.cz, a. s., Lukačovič, I., 2013), and *Jyxo* (CET21, Illich, 2013). When looking at these search engines more closely it can be observed that from the point of view of software engineering aspect they represent a pinnacle of technological achievement in software industry. However, if the point of view of artificial intelligence is adopted then it cannot be certainly said if these search engines are intelligent. For example little is utilized by search engines from series of user's search queries and little support is provided for cooperation among users. It is quite reasonable assumption that a series of queries characterize the effort of what the user want to find better than the single query. The typical search engine however does not help in this effort – users are put into isolation typically which precludes any cooperation and recommendation from other users based on past queries. To be honest, for instance the Bing search engine (more correctly the decision engine) uses certain technology that provide search results based on user's search history and geographical location. This cannot be always said about other search engines. Another relatively weak point of contemporary search engines is that little is done in understanding the search query from the perspective of natural language processing and understanding (Chakrabarti, 2003).

1.1 Through Collaboration to Better Experience: a suggestion

A brief survey of techniques used in contemporary search engines and suggestion of improvements that can enable a better user's experience from the search session are the main parts of the paper. It is suggested to employ cooperation among users in their search effort to achieve the goal. The complete user's interaction with the search engine will be considered – that is whole user's search history as well as the sequence of latest user's queries will be taken into account (Pasca, van Durme, 2007).

1.2 Web Search and Recommender Systems

Recent advances in the area of recommender systems (Ricci, et al., 2011) indicate that great improvement of user's benefit in finding desirable items can be achieved through collaboration among users. Hence it is suggested to integrate techniques of recommender systems with web search.

Until the recent announcement of *Facebook's* (Facebook Inc., 2013) social search engine (*Graph Search*) in January 2013 (Straley, 2013), recommender systems and search engines have been considered mostly separately. Since social networks possesses profiles of millions of users it had a great opportunity to carry out cooperation among users by recommending results to users (or items, products, events, etc...) that have been searched by users similar to them in terms of the profile similarity. It is expected that the attempt will initiate interest in integrations of recommender systems and search engines.

The focus of this work regarding the integration of web search and recommender systems is different than that of used in social networks. It is planned to target research on operators that do not possess any user profiles which is quite rare anyway. The data suggest to be exploited is the user's interaction and behavior. So there is a technical limitation in this sense but on the other hand the limitation represents a research opportunity and also implies wider range of potential operators of such integrated cooperative search service. Certain attempts have been already done in recommending search results using case based reasoning on user's search history (Ross, Wolfram, 2000; Smyth et al., 2011).

2 SURVEY OF THE CURRENT STATE IN WEB SEARCH

The search engine (Büttcher et al., 2010) can be regarded from two different aspects – the software engineering one and the computational intelligence one. From the perspective of software engineering the top level design of traditional search engine is relatively simple; difficulties appear after looking at individual building blocks.

The top level operational design consists of the phase of crawling the web in which textual or multimedia documents are retrieved from the web and stored into the internal storage. Then a phase of indexing follows immediately after. A so called *index* is build in the indexing phase. The index associates searched items, that is words in most cases, with their occurrence in web documents. After the index is built users can post their textual queries through a web interface which are then processed on the server side and searched in the index. Multiple iterations of crawling and indexing are made typically to keep the index up to date while the search service is provided without any interruption.

2.1 Algorithmic and Software Engineering Challenges

One of the most important software engineering challenges appears in building the index. It is very challenging to design the index structure capable of holding the associations of searched items for millions of web documents (consider that *Google* has the index for more than 40e+9 pages; in other major search providers this number is around 20e+9 pages (de Kunder, 2013)) and to keep the speed of important operations such as insertion or search in the index at the acceptable level. This challenge is addressed mostly by efficient programming and by distributing the task. The requirements on the disk capacity to store the index is also a considerable issue. Various techniques of compression are usually employed to make the size of the index manageable (consider that uncompressed index needs many times larger space than the indexed documents themselves).

If no kind of load balancing is considered then the search phase is relatively easy. First, a certain kind of parsing of the user's textual query is made. Then direct searches into the index are performed and results are presented to the user. If too many user's are expected to post queries then the task needs to be distributed on several machines while each of them needs to have access to an up to date index – so certain kind of distribution and/or synchronization of local indexes is imposed by this phase as well.

Let us note that modern algorithmic techniques play a significant role in making the search engines possible. For example modern data structures derived from *trie* (Baeza-Yates, Gonnet, 1996; Fredkin, 1960) such as *suffix trees* (Ukkonen, 1995; Mansour et al., 2011) or *suffix arrays* (Dementiev et al., 2008; Manber, Myers, 1990), are used as building blocks of index structures. Important benefits for user's experience during the search engine session is directly rooted in these data structures – let us mention approximate string matching (Navarro, 2001; Navarro et al., 2001) supported by suffix trees that allows users make typos in their queries.

Another important algorithmic issue that should be in focus of the search engine designer is the *cache* awareness/obliviousness of data structures (Bender et al., 2005) and algorithms (Frigo et al., 2012). As the storage for index must be inherently built in hierarchical manner – from external storage to CPU cache – this issue is a key for the overall performance of the search engine. Moreover classical consideration of cache is reduced on buffering between the main memory and CPU; in the case of index storage the situation is more complicated as there is at least one more layer between main memory and disk. One more layer is represented by the distribution in the case of distributed index.

2.2 Computational Intelligence in Web Search

Into the computational intelligence aspect it is accounted ordering of search results and understanding user's search query. It is well known that a breakthrough in ordering of search results was made by the *PageRank* algorithm (Brin, Page, 1998) that ranks web page by simulating a so called random surfers who randomly follows links in web pages or randomly skips to new web pages. A web page where many random surfers gather are considered to be important and results on them appear on the top. Many other supporting proprietary techniques are used to improve orderings given by *PageRank*. It is necessary to take into account position of searched term within web documents. Another important issue which is however out of scope of this study is security and protection against biasing search results by generating artificial links – the well known spamdexing represents one of these infestations.

The understanding of search query has been addressed in multiple ways. One approach is to analyze the given textual query by techniques of natural language processing which includes syntactic and semantic analysis of the query at the level of sentences (Zhou et al., 2007). The search in the index is performed not only for terms entered by the user but also for terms that are semantically related to them. Semantically related terms are often search in *ontologies*.

An interesting example of syntactic analysis of user queries is represented by the *Sherlock Holmes* search engine (commercially known as *Morfeo*) (Mareš, Špalek, 2009) which performs Czech language stemming of searched terms. Similar techniques are implemented in major search engines mostly for English.

The difficulty of making research in search engines is that it is little publicly known about internal techniques used by major search providers since this represents a proprietary know-how and one of the most valued secrets of each company.

3 A BRIEF SURVEY OF RECOMMENDER SYSTEMS

The second technology in focus of this paper is represented by *recommender systems* (Resnick, Varian, 1997; Ricci et al., 2011). Great advances have been made in automated recommending of various items such as music and movies recently. The most successful technique for making automated recommendations is *collaborative filtering* in which preferences of many users are processed in order to make good recommendations for other users. Knowing what items the user is interested in and preferences of other users, the system is able to recommend other items the user may be also interested in. Contrary to item-based recommendation (Sarwar et al., 2001), the collaborative filtering is able to bring novelty from the perspective of the active user since not only similar items to those already preferred can be recommended (Melville et al., 2002) (if items are represented as vectors then similarity between two items can be calculated as correlation or as their angle, that is, scalar product).

Recommender systems became widely known thanks to commercially successful recommender system of *Amazon.com* (Linden et al., 2003) and thanks to *Netflix prize* (Bell, Koren, 2007). One of the most successful approaches in recommending is a so called *matrix factorization* (Koren et al., 2009.).

If user preferences are represented in a matrix where items corresponds to columns and rows to users then the immediate observation is that this matrix very sparse. The given sparse matrix can be submitted to dimensionality reduction methods (Rennie, Srebro, 2005) that project explicit user preferences into a different space (of smaller dimension) where user preferences are represented in more general categories. If for instance the original matrix represents preferences of particular movies then the transformed one can represent preferences of movie genres. Thus algebraic methods serve here very efficiently for knowledge mining and knowledge discovery.

4 COLLABORATIVE WEB SEARCH

In this preliminary research, it is suggested to integrate techniques from web search with techniques from recommender systems to enable collaborative web search implicitly. Integrating of both approaches also require to develop new techniques in both areas inspired by each other. The implicit cooperation means that the user is not required to do anything else than entering search queries into the search box and picking some of provided results. No acting in sense of the social networking is expected.

4.1 Research Topics in Collaborative Web Search

The top level research question is how to make cooperative recommendations for the user of the web search engine. This means to suggest what the user may be interested in from knowing her/his and other users past short-term and long-term interaction with the search engine. The aspect of novelty in suggesting interesting items should be also investigated since the user may not be sure in what she/he is trying to find and novelty in recommendation could provide her/him new search directions.

It is expected that combination of feature based and semantic methods will be used to make efficient recommendations. Feature based methods include methods that regard the user's preference of searched items as feature vectors and matrices respectively if groups of users are considered, and these collections of features are processed by algebraic and numeric methods such as matrix factorization. The semantic approach includes understanding of searched terms by their meaning in natural language and ability to imply what terms are semantically related. Consider that in the case of search engine there is a weak association between searched terms and items (web documents) the user actually want to find. Hence it is needed to develop techniques that discover these associations automatically. The source of techniques for this task is machine learning (Witten, Frank, 2011).

4.2 Technical Issues in Evaluation

Except this top level research direction there are several minor issues that should be addressed. One of them is evaluation of suggested techniques. Currently it is unclear how to evaluate the search engine in a real-life scenario while scientific attributes of such evaluation are ensured; that is, repeatability and reproducibility of results, since interaction with live users is hard to reproduce. Hence suggestion of relevant test scenarios and acquisition of real life data will be also the task within this research topic. On the other hand web with millions of users represents a huge pool of opportunities to obtain testing data and to make experiments.

Another problematic point that is planned to be addressed is a so called cold start of collaborative recommendation. If no or few data is collected from users of particular recommender system then it is hard to cooperate in suggesting recommendations. One option plan to overcome this problem is using semantic information instead of cooperative one and to gradually increase the cooperative component as the amount of collected data increases.

5 CONCLUSION AND FUTURE WORK

Two areas are in focus of this paper – *web search* and *recommender systems*. A survey of contemporary mainstream search engines is given while certain deficiencies rooted in little cooperation among search engine users are identified. The main suggestion is to improve user's benefit from search engine session by enabling collaboration among users to provide more relevant results. Current techniques used in recommender systems are surveyed and integration of web search with suitable recommender techniques is suggested.

It is planned to integrate recommender techniques into our experimental search engine *yeRCH*. The search engine is implemented in C++, PHP, and JavaScript and all the standard search engine features have been already implemented. A short preliminary demo of the *yeRCH* search engine powering full-text search in an academic web site is shown in the appendix. Data regarding user's behavior are being collected and will be utilized in the future research and development.

ACKNOWLEDGEMENTS

This work is supported by the Charles University in Prague, Czech Republic within the PRVOUK project (section P46) and by the Czech Science Foundation under the contract number GAP103/10/1287.

I would like to thank anonymous reviewers for their critical comments and suggestions that will help me in future research of the topic.

REFERENCES

Linden, G., Smith, B., York, J., 2003. *Amazon.com Recommendations: Item-to-Item Collaborative Filtering.* IEEE Internet Computing, Volume 7 (1), pp. 76-80, http://www.amazon.com/, IEEE Press.

Baeza-Yates, R. A.; Gonnet, G. H., 1996. *Fast text searching for regular expressions or automaton searching on tries.* Journal of the ACM, Volume 43 (6), pp. 915–936, ACM.

Baidu, Inc., 2013. *Baidu Search.* http://www.baidu.com/, China, (Accessed on March 2013).

Bell, R. M., Koren, Y., 2007. *Lessons from the Netflix Prize Challenge.* SIGKDD Explorations, Volume 9, pp. 75-79, ACM.

Bender, M. A., Demaine, E. D., Farach-Colton, M., 2005. *Cache-Oblivious B-Trees*. SIAM Journal of Computing, Volume 35(2), pp. 341-358, ACM.

Brin, S., Page, L., 1998. *The anatomy of a large-scale hypertextual Web search engine.* Computer Networks and ISDN Systems, Volume 30, pp. 107–117, Elsevier.

Büttcher, S., Clarke, C. L. A., Cormack, G., V., 2010. *Information Retrieval: Implementing and Evaluating Search Engines.* MIT Press.

CET21, Illich, M., 2013. *Jyxo search / Yoopy.* http://sluzby.yoopy.cz/, Czech Republic, (Accessed on March 2013).

Chakrabarti, S., 2003. *Mining the web - discovering knowledge from hypertext data*, pp. I-XVIII, 1-345, Morgan Kaufmann.

Dementiev, R., Kärkkäinen, J., Mehnert, J., Sanders, P., 2008. *Better external memory suffix array construction.* ACM Journal of Experimental Algorithmics, Volume 12, ACM.

Facebook Inc., 2013. *facebook - Connect with friends and the world around you on Facebook.* http://www.facebook.com, USA, (Accessed on March 2013).

Fredkin, E., 1960. *Trie Memory.* Communications of the ACM, Volume 3 (9), pp. 490–499, ACM.

Frigo, M., Leiserson, C. E., Prokop, H., Ramachandran, S., 2012. *Cache-Oblivious Algorithms.* ACM Transactions on Algorithms, Volume 8(1), ACM.

Google Inc., 2013. *Google Search.* http://www.google.com/, USA, (Accessed on March 2013).

Koren, Y., Bell, R. M., Volinsky, C., 2009. *Matrix Factorization Techniques for Recommender Systems.* IEEE Computer, Volume 42 (8), pp. 30-37, IEEE Press.

de Kunder, M., 2013. *The size of the World Wide Web.* http://www.worldwidewebsize.com/, Netherlands, (Accessed on March 2013).

Manber, U., Myers, G., 1990. *Suffix arrays: a new method for on-line string searches.* Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms, pp. 319-327, ACM.

Mansour, E., Allam, A., Skiadopoulos, S., Kalnis, P., 2011. *ERA: Efficient Serial and Parallel Suffix Tree Construction for Very Long Strings.* Proceedings of the VLDB Endowment, Volume 5 (1), pp. 49–60, University of Michigan.

Mareš, M., Špalek, R., 2009. *Sherlock Holmes Search Engine*. http://www.ucw.cz/holmes/, Czech Republic, (Accessed on March 2013).

Melville, P., Mooney, R. J., Nagarajan, R., 2002. *Content-Boosted Collaborative Filtering for Improved Recommendations.* Proceedings of the 18th National Conference on Artificial Intelligence (AAAI), pp. 187-192, AAAI Press.

Microsoft Corp., 2013. *Bing Search.* http://www.bing.com, USA, (Accessed on March 2013).

Navarro, G., 2001. *A guided tour to approximate string matching.* ACM Computing Surveys, Volume 33 (1), pp. 31–88, ACM, 2001.

Navarro, G., Baeza-Yates, R. A., Sutinen, E., Tarhio, J., 2001. *Indexing Methods for Approximate String Matching.* IEEE Data Engineering Bulletin 24 (4): pp. 19–27, IEEE Press.

NHN Corp., 2013. *Naver Search.* http://www.naver.com/, South Korea, (Accessed on March 2013).

Pasca, M., van Durme, B., 2007. *What you seek is what you get: Extraction of class attributes from query logs.* Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), pp. 2832-2837, IJCAI, 2007.

Resnick, P., Varian, H., 1997. *Recommender systems.* Communications of the ACM, Volume 40 (3), pp. 56–58, ACM.

Rennie, J. D. M., Srebro, N., 2005. *Fast Maximum Margin Matrix Factorization for Collaborative Prediction*. Machine Learning, Proceedings of the 22nd International Conference (ICML 2005), pp. 713-719, ACM International Conference Proceeding Series 119, ACM.

Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Editors), 2011. *Recommender Systems Handbook*. Springer Verlag.

Ross, N., Wolfram, D., 2000. *End user searching on the Internet: An analysis of term pair topics submitted to the Excite search engine.* Journal of the American Society for Information Science, Volume 51 (10), pp. 949–958, JASIST.

Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J., 2001. *Item-based collaborative filtering recommendation algorithms.* Proceedings of the 10th International World Wide Web Conference (WWW 2001), pp. 285-295, ACM.

Seznam.cz, a. s., Lukačovič, I., 2013. *Seznam search.* http://www.seznam.cz/, Czech Republic, (Accessed on March 2013).

Smyth, B., Freyne, J., Coyle, M., Briggs, P., 2011. *Recommendation as Collaboration in Web Search.* AI Magazine, Volume 32(3), pp. 35-45, AAAI Press.

Smyth, B., Coyle, M., Briggs, P., 2012. *HeyStaks: a real-world deployment of social search.* Proceedings of Sixth ACM Conference on Recommender Systems (RecSys 2012), http://www.heystaks.com/, pp. 289-292, ACM, (Accessed on March 2013).

Straley, B., 2013. Facebook's Graph Search: the Ultimate Personalized Discovery Engine? http://searchenginewatch.com/article/2238590/Faceboks-Graph-Search-the-Ultimate-Personalized-Discovery-Engine, Search Engine Watch, January 23, 2013, (Accessed on March 2013).

Ukkonen, E., 1995. *On-line construction of suffix trees.* Algorithmica, Volume 14 (3), pp. 249–260, Springer Verlag.

Witten, I. H., Frank, E., 2011. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

Yahoo! Inc., 2013. *Yahoo! Search*. http://www.yahoo.com/, USA, (Accessed on March 2013).

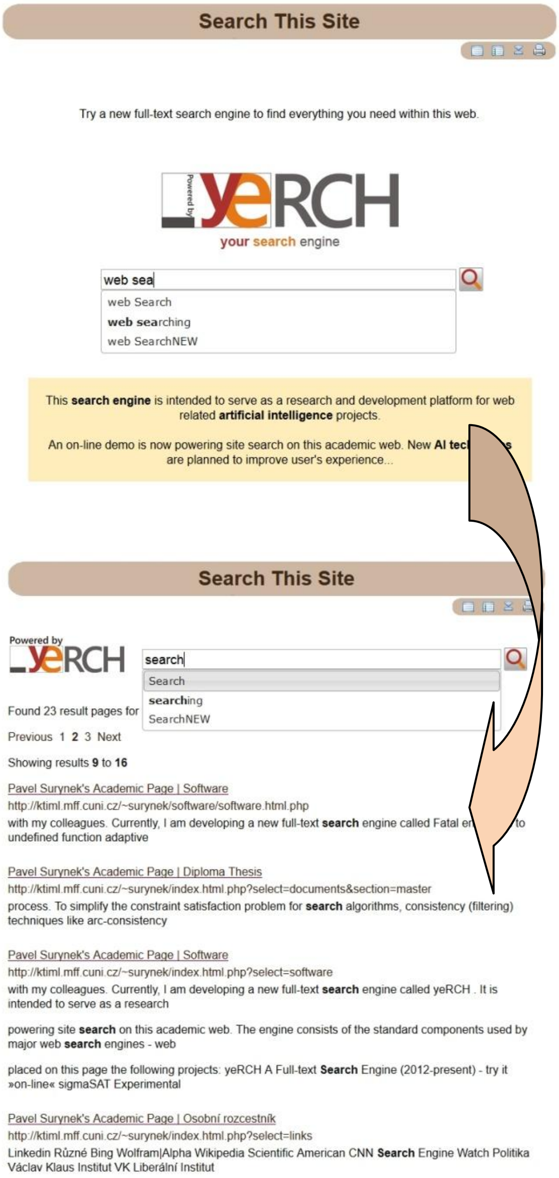
Yandex Corp., 2013. Yandex Search. http://www.yandex.ru/, Russia, (Accessed on March 2013).

Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y., 2007. *Spark: adapting keyword query to semantic search.* The Semantic Web, pp. 694-707, Springer Verlag, 2007.

APPENDIX – YERCH SEARCH ENGINE EXPERIMENTS

A prototype search engine called *yeRCH* has been developing to be able to conduct experiments with cooperation in web search. *yeRCH* search engine is now powering full-text search on the academic web site of the author's institute. Data collected from the search engine operation will be further processed in order to improve users experience (entered search terms, clicked through results, etc. are collected).

An initial experiment has been done with the search engine. If results are summarized, the most important observation from collected users behaviour is that users often use the search to find concrete term on the web while few related searches are invoked. Hence it is considered that adding recommendation of related searches would increase user's benefit from the search.



Try a new full-text search engine to find everything you need within this web.

This **search engine** is intended to serve as a research and development platform for web related **artificial intelligence** projects.

An on-line demo is now powering site search on this academic web. New **AI tec s** are planned to improve user's experience...

Found 23 result pages for

Previous 1 **2** 3 Next

Showing results **9** to **16**

Pavel Surynek's Academic Page | Software
http://ktiml.mff.cuni.cz/~surynek/software/software.html.php
with my colleagues. Currently, I am developing a new full-text **search** engine called Fatal en to undefined function adaptive

Pavel Surynek's Academic Page | Diploma Thesis
http://ktiml.mff.cuni.cz/~surynek/index.html.php?select=documents&section=master
process. To simplify the constraint satisfaction problem for **search** algorithms, consistency (filtering) techniques like arc-consistency

Pavel Surynek's Academic Page | Software
http://ktiml.mff.cuni.cz/~surynek/index.html.php?select=software
with my colleagues. Currently, I am developing a new full-text **search** engine called yeRCH . It is intended to serve as a research

powering site **search** on this academic web. The engine consists of the standard components used by major web **search** engines - web

placed on this page the following projects: yeRCH A Full-text **Search** Engine (2012-present) - try it »on-line« sigmaSAT Experimental

Pavel Surynek's Academic Page | Osobní rozcestník
http://ktiml.mff.cuni.cz/~surynek/index.html.php?select=links
Linkedin Různé Bing Wolfram|Alpha Wikipedia Scientific American CNN **Search** Engine Watch Politika Václav Klaus Institut VK Liberální Institut

Figure 1: Illustration of using *yeRCH* search engine powering full text search on the author's academic web site.