An Application of Personalized PageRank Vectors: Personalized Search Engine

Mehmet S. Aktas[1,2], Mehmet A. Nacar[1,2], and Filippo Menczer[1,3]

[1] Indiana University, Computer Science Department
Lindley Hall 215 150 S. Woodlawn Ave.
Bloomington, IN 47405-7104
{maktas, mnacar, fil}@indiana.edu
http://www.cs.indiana.edu
[2] Indiana University, Community Grids Labs
501 N. Morton St. Room 222 Bloomington, IN 47404
http://www.communitygrids.iu.edu
[3] Indiana University, Informatics School
901 East 10th Street Bloomington, IN 47408-3912
http://www.informatics.indiana.edu

Abstract. We introduce a tool which is an application of personalized pagerank vectors such as personalized search engines. We use pre-computed pagerank vectors to rank the search results in favor of user preferences. We describe the design and architecture of our tool. By using pre-computed personalized pagerank vectors we generate search results biased to user preferences such as top-level domain and regional preferences. We conduct a user study to evaluate search results of three different ranking methods such as similarity-based ranking, plain PageRank and weighted (personalized) PageRank ranking methods. We discuss the results of our user study and evaluate the benefits our personalized PageRank vectors in personalized search engines.

1 Introduction

The Web is a highly distributed and heterogeneous information environment. The immense number of documents on the Web produces various challenges for search engines. Storage space, crawling speed, computational speed and retrieval of most relevant documents are some examples of these challenges. Intuitively, given a query, most relevant documents can be considered as the most authoritative documents that match with that query. Recent information retrieval techniques, such as PageRank [1],[2] and HITS [3], puts together the traditional similarity matching retrieval method with a notion of popularity of links, based on the hypertext structure of the Web.

PageRank algorithm provides a global ranking of the web pages based on their importance ([1],[4],[15]). For instance, a link from a web page "A" to another web page "B" can be considered as if page "A" is voting for the importance of page "B". So, as the number of in-links of page "B" gets increased, its importance gets increased as well. PageRank also considers the importance of in-links. Not only the number of in-links but also the importance of these in-links decides PageRank of a page. In this scenario, global ranking of the pages is based on the web's graph structure. Search engines, such as Google[8], utilizes the link structure of the Web to calculate PageRank values of the pages. Then, these values are used to re-rank search results to improve precision. There are comprehensive surveys published in explaining the issues related with PageRank in [5][6][7].

The importance of web pages for different users can be better determined, if the PageRank algorithm takes into consideration user preferences. Personalized PageRank approach was first introduced in [4] and studied by others [10][11] as query-dependent ranking mechanism. However, the use and benefits of personalized PageRank vectors have not been studied and explored enough in personalized web search applications.

In this paper, we introduce a personalized search engine that utilizes personalized PageRank vectors. We study PageRank ranking method focusing particularly on personalized PageRank vectors. We define the notion of relevancy of documents as a subjective metric which depends heavily on user satisfaction. We emphasize that preferences should play an important role in calculating the PageRank values. We implement a personalized search engine as an application of personalized PageRank vectors. We calculate PageRank vectors offline prior to search by taking into consideration of the personal preferences of the users. Our aim is to further improve the precision at low recall. We describe the design and architecture of our application in details. We also explore the improvement we gain in precision by using personalized PageRank vectors. In the next section we talk about motivations that led us to do this research.

1.1. Motivations

PageRank algorithm provides an objective view of the web when deciding the global importance of web pages. However, such objectivity brings various problems.
In some domains, highly ranked and authoritative web pages might be distributed into various regions, such as Europe, America and Asia. Plain PageRank algorithm [1][4] does not take into consideration the regional choices of users. So, the search results might also include sites from regions that a user might have the least interest. However, a user would be much more interested in the sites that are highly ranked and that are in the same region as user.

An important consideration of the web users is the reliability of information available on the web. The web has a democratic structure, in other words, majority of the web sites on the internet are not monitored for the information that are published. To this end, it is important to choose the information sources that are more likely to be monitored by experts for the information accuracy and the quality. For instance, a user might favor to the sites that are on education top-level domain since they tend to reflect the point of view of highly educated community. Likewise, another user might favor government pages, since they tend to be monitored more strictly compared to other domains for information accuracy.

So, some web pages, with many high ranked in-links from a top-level domain might appear as if they are the most qualified and authoritative information sources in some topics, whereas they maybe not very well monitored for the information accuracy and quality. To this end, search engines might return pages that might not give information satisfying user needs and preferences.

The importance of a page differs for different individuals with different interests, knowledge and background. So, a global ranking of a web page might not necessarily indicate the importance of that page for individual users. It is important to calculate personalized view of importance of the pages. In order to overcome these problems, we introduce a method which is based on calculating the personalized PageRank vectors prior to query time.

The outline of the paper is as follows. First we talk about the use of personalized PageRank vectors in our application. Second, we explain the design and architecture of our personalized search engine. Third, we discuss the experiments that we applied to explore the improvements when using personalized PageRank vectors. Fourth, we present and discuss our results. At last, we finalize our paper with a conclusion.

In the following section, we discuss how we use personalized PageRank vectors in our search engine in details.

2. Personalized PageRank Vectors

Personalized PageRank vectors provide a ranking mechanism which in turn creates a personalized view of the web for individual users. An example application of personalized PageRank vectors could be personalized search engines. In this section, we discuss the use of personalized PageRank vectors in our implementation of personalized search engine.

The computation of personalized PageRank vectors are done prior to search time. When calculating the PageRank vectors, pre-defined user profiles are taken into consideration. We use following equations in Figure 1 in order to calculate both plain and personalized PageRank scores.

Definitions:

$\mathrm{Pr_p}(\mathrm{A})$ = Plain PageRank score of a page A.

$\mathrm{Pr_w}(\mathrm{A})$ = Weighted (personalized) PageRank score of a page A.

T_i = i th parent of a page A.

d = dumping factor

$w(T_i)$ = normalized weight factor computing by applying links analysis on parent page T_i.

$\mathrm{C}(T_i)$ = number of out-links of parent page T_i.

$$\mathrm{Pr_p}(\mathrm{A}) = (1-d) + d \times \left(\frac{\mathrm{Pr}(T_1)}{\mathrm{C}(T_1)} + \ldots + \frac{\mathrm{Pr}(T_n)}{\mathrm{C}(T_n)} \right)$$

$$\mathrm{Pr_w}(\mathrm{A}) = (1-d) + d \times \left(w(T_1)\frac{\mathrm{Pr}(T_1)}{\mathrm{C}(T_1)} + \ldots + w(T_n)\frac{\mathrm{Pr}(T_n)}{\mathrm{C}(T_n)} \right)$$

Fig.1. Plain and weighted PageRank equations used in our experiments

In our design, a user profile consists of six different top level domains and three different regions as choices of user preferences. The top-level domains are commercial (.com), military (.mil), government (.gov), organization (.org), business (.net) and education (.edu) domains. We also introduce following regions as choices for regional preferences; Asia, America and Europe. To this end, in order to calculate personalized PageRank scores, we calculated 2^9 = 512 different combinations of user preference choices. So, we pre-computed 512 different user profiles.

An important feature our approach is that we apply link analysis to a URL of a web page to compute its personalized PageRank score. After an analysis of its URL, a web page might be classified as if it belongs to a top-level domain, a region, both or none of them. So, based on this analysis, a URL will get a weight factor. There are 27 possible weight factors for each user profile. These weight factors can be summarized as follows. A URL might belong to only one of the top-level domain (out of 6 top-level domains). A URL might belong to only one of the region (out of 3 possible regions). A URL might belong to both a region and a top-level domain at the same time (out of 18 different combinations of (top-level domain, region) pairs). The values of these weight factors will vary for each user profile. If a page does not qualify for any of the possible weight factor category, then plain PageRank score is calculated. We can illustrate this with following example.

Example: A government site belongs to United Kingdom: “http://www.direct.gov.uk”
A pre-defined user profile: region: America, Europe
top-level domain: government, education

In this example, a personalized PageRank score for a government site belonging to United Kingdom can be computed as following. We analyze the given url by looking at its top-level domain and country extension. We simply do this by examining its anchor text and compare the result of this examination with our database where we store all possible top-level domain abbreviations as well as abbreviations for country extensions. Since we already know which country located at which region, by finding the country extention, we simply look up for the corresponding region.

In this example, “http://www.direct.gov.uk” belongs to region Europe and top-level domain government. Since the given url happens to have both top-level domain and region that exist in the given user profile, it gets the highest normalized weight factor which is “1” in this example. Table-1 shows the weight correlation for the example above.

Table 1. Following table is a weight correlation table for pre-defined user profile with following preferences. Region preferences: America, Europe, top-level domain preferences: education, government.

combinations of top-level domains and regions	weight factors	normalized weight factors
education	**1**	**0.5**
europe	**2**	**0.5**
government	**2**	**0.25**
.	.	.
europe & government	**4**	**1**
.	.	.

A user is expected to input his/her choices of interests before the query time. When a query is posed by a user, based on his/her user profile, we retrieve the corresponding personalized PageRank vector in order to re-rank the hits to satisfy the query. We multiply the TFIDF based similarity score with PageRank scores. Resulting scores form the final ranking score to re-rank the hits. By multiplying the similarity based metric with PageRank scores we aim to increase precision at low recall for our implementation of personalized search engine.

In the following section we discuss the details of our design and architecture of our implementation in details.

3. Design and Architecture

We designed our personalized search engine as a Java program that utilizes Nutch project [14] as a main search engine which use TFIDF bases similarity metric. The implementation consists of two core parts.

First part of the implementation focus on calculations that happen prior to search time. In this part, we pre-compute a number of personalized PageRank vectors as well as a plain PageRank vector. Personalized PageRank vectors are computed based on pre-defined user profiles. User profiles include choices of user interests in region and/or top-level domains of web pages. PageRank vectors are computed only once prior to query time and stored as data files. In order not to increase the online query time, we also implemented an extensions to Nutch project index system, so that it can also accommodate PageRank scores along with the existing information such as anchor text, keywords, and similarity score. This avoids the heavy I/O overhead of reading the PageRank results from a file store or a database. The computation of the PageRank vectors happen after the creation of a connected web graph. Data structure of the web graph is an important part of our implementation. We used compressed sparse row (CSR) data structure for adjacency matrix for data representation. CSR data sturucture stores its row and column index for each entry. Entries are listed one row after another. This is simply done by defining a data structure which is a triplet (i,j,value).

For this we defined a java object to represent a triplet and a global array to store the triplet objects. By doing this, we don't store non-zero values unnecessarily.

We used global parallel arrays for vertices and PageRank vectors. After PageRank vectors are computed, each PageRank vector is dumped into an output file. Prior to calculation of personalized PageRank results, we created weight correlation Java Objects for each possible user profile. In our design, we presented 9 different choice of interest in top level domain and regions as mentioned before. This is why we created 2^9 different weight correlation objects each is corresponding to a different user profile. Each weight correlation object includes 27 different weight factor for different user preference combinations of top level domains and regions. When calculating the PageRank of a page a link analysis is done. Based on the result of this analysis, we determine which url belongs to which top level domain and/or region. We used a hash table within the weight correlation objects to store user preference combinations. After the PageRank results are computed and stored in a file store, we run our extended version of Nutch index system to store newly created PageRank scores in Nutch index database.

Second part of our implementation focus on online query processing and our user study. In this part, we implemented various user interfaces by using Java Server Pages. When a query is made, we use Nutch searcher mechanism to retrieve the hits. Nutch returns a TFIDF based similarty score for each hit. We re-order the hits based on plain PageRank and personalized PageRank scores. We use three global arrays to store the ranking scores of the hits that belong to three different ranking mechanism such as similarity based, plain PageRank and personalized PageRank ranking mechanisms. In order to include the similarity based metric into PageRank ranking methods, we calculate final plain and personal PageRank scores by multiplying similarity based Nutch score with pre-calculated PageRank values.

So far, we have discussed the details of personalized PageRank vectors in our implementation and design and the architecture of our personalized search engine. We discuss the details of the experiments and our user study in the following section in details.

4 Experiments

We conducted a user study to measure the performance of different ranking methods such as similarity-based, plain PageRank and weighted PageRank (personalized) ranking methods. In this study, we asked each volunteer to use our personalized search facility after they input their user profiles into our system. After making a query, each volunteer was shown search results from three different ranking mechanism. For each query, top 10 results from each ranking method are considered and these results are randomly shuffled before they were shown to volunteers. There are 5 human subjects who contributed to our user study in total with 10 queries. We realize that recall and precision values are dependent on whether the human subjects of a user study are experienced searchers or not. An experienced searcher may effect the recall and precision, either by finding everything on a topic or only few relevant results. To this end, we conducted our user study with a group of graduate students and did not give out any information about the main goal of the search engine. Volunteers are only expected to select relevant Urls that satisfies their choice of preferences as well.

In order better explain our method in the user study, we would like to explain it with an example. Suppose that, Nutch search engine returns at least 10 results satisfying a query. Then, these results are re-ranked based on two other PageRank based ranking methods. If top 10 hits from three different ranking mechanisms turn out to be totally different from each other, then the volunteer was shown 30 hits as a result of his/her query. If the hits from different ranking mechanisms overlap with each other, then the number of results will range from 10 to 30.

After the results are shown, each user was asked to select the URLs that are relevant to his/her query as well as satisfying his/her choice of preferences. Let a user is shown 30 results satisfying his/her query. Also, let this user select 8 out of 30 results as relevant. In this case, for each ranking mechanism, the recall is the division of the number of relevant URLs coming from corresponding ranking mechanism by the total number of URLs deemed to be as relevant which is 8. Likewise, for each ranking mechanism, the precision is the division of the relevant URLs coming from corresponding ranking mechanism by the number of retrieved results which is 10 in our example.

To this end, when a query is made, we calculate (precision, recall) pairs for each ranking mechanism. Also, the average of all (precision, recall) pairs are calculated for all given queries. The definitions of the parameters and the calculations used in calculating the (precision, recall) pairs are shown below.

Definitions:

R = total set of the ranking mechanisms
r = type of the ranking mechanism
q = query
i = position of the URL in the combined hit list, starting from first URL in the list to the last, from top to bottom.

$retrieved(r,i,q)$ = number of all retrieved documents for ranking mechanism r, and query q

$relevant(r,i,q)$ = number of retrieved documents that are deemed as relevant by evaluating the URLs

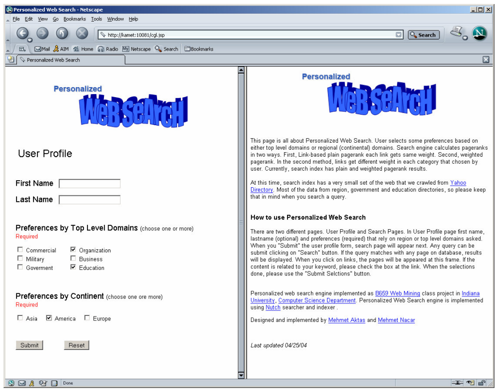
$$precision(r,i,q) = \frac{relevant(r,i,q)}{\bigcup_{i \in Q} retrieved(r,i,q)}$$

$$recall(r,i,q) = \frac{relevant(r,i,q)}{\bigcup_{r \in R} \bigcup_{i \in Q} relevant(r,i,q)}$$

Fig.2. Definition of precision and recall formulas that we used in our user study

For the experiments, we used a crawl data that we gained by crawling the web with the start point of Yahoo Directory [9] sub-categories such as “Education”, “Region” and “Government” in April 2004. As a result, our data consists of 107890 Urls and 468410 edges to connect them in a web graph. The dynamic nature and the growth of the web makes it difficult to calculate connectivity based ranking mechanism scores such as PageRank ranking score. When calculating the PageRank scores, the problem of danglink links, nodes that don’t have known outlinks, was explained in [4]. Also, it has been showed that it is possible to compute PageRank scores under the effect of missing outlink information and keeping the PageRank errors under control in [15]. In our experience, in order to avoid the high error rate in PageRank calculations and increase the size of our data, we used an additional imaginary node to distribute the PageRank from danglink links back to the graph, by connecting danglink links with source nodes, nodes that don’t have known inlinks, through an imaginary node. We experimentally observed that the order of the pages change with a negligible error rate, when using an imaginary node to distribute the rank from danglink links back to the graph.

For the user study, we designed easy-to-use user interfaces. We aim to reduce possible wrong evaluations that might be caused by complicated user interfaces. Our user study interfaces have a modular structure and it is flexible for modifications. It consists of three parts. In the first part, users provide personal information such as first name, last name and choices of interests in top-level domains and regions. First part of our user interfaces is illustrated in Figure-4. Second part of the user study is where the personalized web search facility is displayed. Users are expected to pose their queries by using this search facility. The third part of the user study interfaces is where the top hits of three different ranking mechanisms are displayed to the user. In this part, we also provide facilities like navigating the hits and selecting relevant pages that satisfy the user query. The third part of our user interface is shown in Figure-5.


Personalized Web Search - Netscape
File Edit View Go Bookmarks Tools Window Help
http://kamet:10081/cgl.jsp
Search
Mail AIM Home Radio Netscape Search Bookmarks
Personalized Web Search
Personalized
WeB SeArcH
Done

User Profile

First Name

Last Name

Preferences by Top Level Domains (choose one or more)
Required

☐ Commercial ☑ Organization
☐ Military ☐ Business
☐ Goverment ☑ Education

Preferences by Continent (choose one ore more)
Required

☐ Asia ☑ America ☐ Europe

Submit Reset

This page is all about Personalized Web Search. User selects some preferences based on either top level domains or regional (continental) domains. Search engine calculates pageranks in two ways. First, Link-based plain pagerank each link gets same weight. Second, weighted pagerank. In the second method, links get different weight in each category that chosen by user. Currently, search index has plain and weighted pagerank results.

At this time, search index has a very small set of the web that we crawled from Yahoo Directory. Most of the data from region, government and education directories, so please keep that in mind when you search a query.

How to use Personalized Web Search

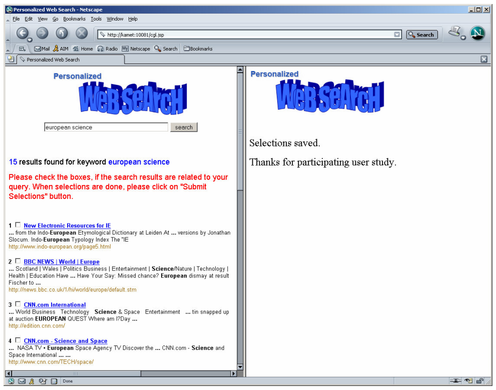
There are two different pages. User Profile and Search Pages. In User Profile page first name, lastname (optional) and preferences (required) that rely on region or top level domains asked. When you "Submit" the user profile form, search page will appear next. Any query can be submit clicking on "Search" button. If the query matches with any page on database, results will be displayed. When you click on links, the pages will be appeared at this frame. If the content is related to your keyword, please check the box at the link. When the selections done, please use the "Submit Selctions" button.

Personalized web search engine implemented as B659 Web Mining class project in Indiana University, Computer Science Department. Personalized Web Search engine is implemented using Nutch searcher and indexer .

Designed and implemented by Mehmet Aktas and Mehmet Nacar

Last updated 04/25/04

Fig. 4. *User Profile Page.* User enters his/her firs name, last name and choices of interests of top level domains and/or region to create a user profile.


Personalized Web Search - Netscape
File Edit View Go Bookmarks Tools Window Help
http://kamet:10081/cgl.jsp
Search
Mail AIM Home Radio Netscape Search Bookmarks
Personalized Web Search
Personalized
WeB SeArcH
search
Done

european science

15 results found for keyword european science

Please check the boxes, if the search results are related to your query. When selections are done, please click on "Submit Selections" button.

1 □ New Electronic Resources for IE
... from the Indo-**European** Etymological Dictionary at Leiden At ... versions by Jonathan Slocum. Indo-**European** Typology Index The "IE
http://www.indo-european.org/page5.html

2 □ BBC NEWS | World | Europe
... Scotland | Wales | Politics Business | Entertainment | **Science**/Nature | Technology | Health | Education Have ... Have Your Say: Missed chance? **European** dismay at result Fischer to ...
http://news.bbc.co.uk/1/hi/world/europe/default.stm

3 □ CNN.com International
... World Business Technology **Science** & Space Entertainment ... tin snapped up at auction **EUROPEAN** QUEST Where am I?Day ...
http://edition.cnn.com/

4 □ CNN.com - Science and Space
... NASA TV • **European** Space Agency TV Discover the ... CNN.com - **Science** and Space International ... ...
http://www.cnn.com/TECH/space/

Selections saved.

Thanks for participating user study.

Fig. 5. *Personalized Web Search Page.* User makes a query by using personalizes web search facility. Search results coming from different ranking methods are shuffled. User enters his/her choices regarding relevant results without knowing which result belong which search method. User selections are saved after each query.

5 Results

The precision and recall values are summarized by the plot in Figure 6 below. These values are gathered by averaging (precision, recall) pairs of all queries posed by the users. In order to emphasize the difference in precision of ranking mechanisms, we applied the logarithmic interpolation to each line of the graph.

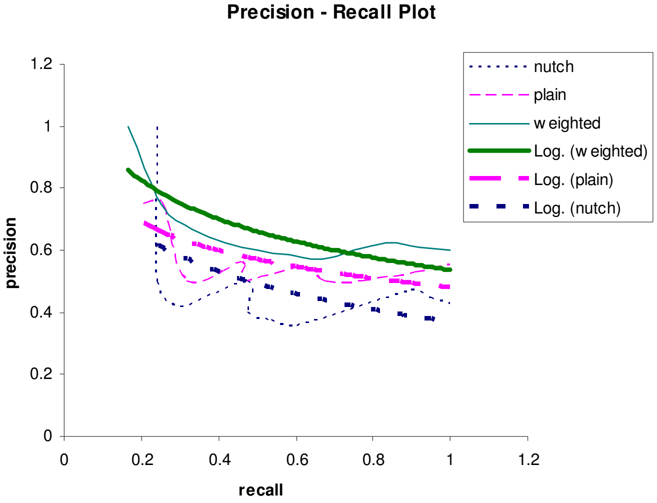


Fig. 6. Comparison of precision and recall plot of three different ranking mechanisms.

The most significant conclusion we can draw from the plot is that personalized PageRank vectors provide better precision than other two ranking mechanisms. We can also see that both PageRank based ranking methods outperforms similarity based ranking method by providing better precision. Based on these results, we manage to provide personalized view of importance of web pages and a better ranking mechanism in our personalized web search engine.

6 Relevant Work

Personalized search engines were first introduced as an application of personalized PageRank vectors in [4], however, newer explored. There are 2^n different personalized PageRank vectors for all possible user preferences. This requires enourmous amount of computation and storage facilities. In attempt to solve this problem, a method was introduced in [11] that computes only limited amount of PageRank vectors offline. This method suggests usage of partially-computed PageRank vectors that are practical to compute and store offline. Based on this method, personalized PageRank vectors for other possible user preferences can be computed fast at a query time. In our work, our main concern is to introduce a prototype personalized search engine, where we can apply experiments to find out the improvement in precision when using personalized PageRank vectors. To this end, we limit the range of the user preferences to top-level domain preferences and regional preferences. Since the total number of preference sets were reasonable to compute and store offline, we do not need calculating the personalized PageRank vectors at query time.

There has been extensive research done on calculating PageRank scores in efficient ways with various techniques [16][17][18][19]. Since our main focus was to utilize personalized PageRank vectors in personalized web search engines and explore the improvements, we only implemented a prototype where we did not expect it to scale up to the size of the current web. Also, we did not anticipate to calculate the pagerank vectors with frequent intervals. This is why, we did not use these efficient ways of calculating pagerank vectors in our implementation. The computation time overhead for limited number of personalized PageRank vectors was manageable for our purposes of research.

“Topic-sensitive” web search, introduced in [10], and the intelligent random surfer, introduced in [13], are similar to our work. Both methods suggest pre-computation of personalized PageRank vectors prior to query time and calculation of PageRank vectors based on query similarity. When a query is made, corresponding personalized PageRank vector is selected according to the similarity between query and the topic of the PageRank vector. Our work is similar, since we also pre-compute personalized PageRank vectors and then use the corresponding PageRank vector to re-order the hits at query time. However, our main difference is that we perform link analysis such as top-level domain analysis and/or reginal extention analysis, when calculating and selecting corresponding personalized PageRank vectors. Our method does not require the full content of a page, since we are only interested in anchor text of the URLs when deciding on the weight correlations.

7 Conclusions

In this paper, we introduced a tool which is a personalized web search engine as an applicaton of personalized PageRank vectors. First, we discussed our use of personalized PageRank vectors in our implementation. Apart from similar research in the area, we focused on the link analysis of the URLs such as top-level domain and regional extension analysis, when computing the personalized PageRank vectors. Then, we explained our design and architecture in implementing the system. We designed and conducted a user study. In the user study, we ran experiments on a real crawl data to explore improvements in precision when user preferences are taken into consideration in PageRank calculation. At last, we presented the results that shows improvement in precision when using personalized PageRank vectors.

We conclude that URL-analysis-dependent personalized PageRank scores can provide higher quality of search results and better precision at low recall. In the future work, we plan to explore efficient ways of calculating PageRank scores in order to enable our personalized search engine scale upto dynamic nature and the growth of the web.

References

1. “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Sergey Brin and Law rence Page, http://www-db.stanford.edu/~backrub/google.html
2. “What can you do with a Web in your Pocket”, Sergey Brin, Rajeev Motwani, Larry Page and Terry Winograd, In Buletin of the IEEE Computer Society Technical Committee on Data Engineering, 1998
3. “Authoritative sources in a hyperlinked environment”, J. M. Kleinberg, Proceedings of the Ninth Annual ACM-SIA Symposium on Discrete Algorithms, 1998
4. “The PageRank Citation Ranking: Bringing Order to the Web” by Lawrence Page, Sergey Brin, Rajeev Motwani und Terry Winograd, Technical report, Stanford University Database Group, 1998
5. “Searching the web”, Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paep cke, and Sriram Raghavan. ACM Transactions on Internet Technology, 2001.
6. “A survey of eigenvector methods of web information retrieval” Amy N. Langville and Carl D. Meyer. The SIAM Review, 2003. Accepted in December 2003.
7. “Deeper inside pagerank”, Amy N. Langville and Carl D. Meyer. Internet Mathematics Journal, 2004. Accepted in February 2004.
8. Google Web Site: http://www.google.com
9. Yahoo Directory Web Site: http://dir.yahoo.com
10. “Topic-sensitive PageRank “, T. H. Haveliwala. In Proceedings of the Eleventh Interna tional World Wide Web Conference, Honolulu, Hawaii, May 2002.
11. “Scaling personalized websearch” G. Jeh and J. Widom, Technical report, Stanford Univer sity Database Group, 2002
12. “An Analytical Comparison of Approaches to Personalizing PageRank”, Haveliwala T., Kamvar S., Jeh G. Stanford University Technical Report, 2003.
13. “The intelligent surfer: Probabilistic combination of link and content information in PageRank”, M. Richardson and P. Domingos, In Proceedings of Advances in Neural Information Processing Systems 14, Cambridge, Massachusetts, December 2002

14. Nutch Open Source Search Engine Site: http://www.nutch.org
15. “Outlink Estimation for Pagerank Computation under Missing Data”, Acharyya S., Ghosh J., The Thirteenth International World Wide Web Conference, N.Y. 2004
16. “Efficient computation of PageRank”, Taher H. Haveliwala, Stanford University Technical Report, 1999
17. “Exploiting the block structure of the web for computing PageRank”. Kamvar S. D., Haveliwala T. H., Manning C. D., and Golub G. H.. Stanford University Technical Report, 2003.
18. “Extrapolation Methods for Accelerating PageRank Computations”, Kamvar S., Haveliwala T., Manning C., Golub G., Proceedings of the Twelfth International World Wide Web Confer ence, 2003.
19. “Adaptive Methods for the Computation of PageRank”, Kamvar S., Haveliwala T., and Golub G., Technical report, Stanford University, April 2003.