Personalized Web Search

Mehmet S. Aktas and Mehmet A. Nacar
Computer Science Department
Indiana University, Bloomington, IN

Abstract:

We introduce personalized PageRank vectors to improve PageRank ranking method. We include the user preferences into calculation of the PageRank. We calculate previously built PageRank vectors to rank the search results based on user preferences. We conduct a user study to find out if our approach provides a better ranking mechanism.

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. In this picture, it is important to define the relevancy of the documents as most popular and best quality documents. When ranking the html pages, you may judge about the quality of a page: by analyzing its content, by measuring its popularity or by examining its connectivity.

In this paper, we focus on connectivity based quality metric, PageRanking, to improve the information retrieval quality. Such connectivity-based metrics do not require retrieving full page content. They only require retrieving the links on a page. Pagerank defines the important of a page by calculating the weighted sum of the back links to it in a recursive way. PageRank provides high precision at the expense of low recall. Global ranking of the pages is based on the web's graph structure. The idea of PageRank relies on the intuition that counting backlinks of a web page gives an indication about the importance of that page. Search engines, such as Google, 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 the precision. PageRank algorithm has been studied in various aspects in hypertext community in recent years [1][2][3][4].

In this paper, we introduce personalized PageRank vectors to further improve the search results and precision. We define the notion of relevancy of documents as a subjective metric which depends heavily on user satisfaction. In this scenario, user's preferences play an important role in calculating the PageRank values. We implemented our idea of personalized PageRank vectors to explore the improvements in the search results. In the following sections of this paper we discuss the details of our idea and the details of the implementation.

The outline of the paper is as follows. First we talk about the implementation and architecture of our system in details. Second, we explain the user interfaces. Third, we discuss the integration of our implementation with Nutch [5] search engine. Fourth, we

discuss our user evaluations and the results. At last, we finalize our paper with a conclusion.

Architecture and Implementation

The PageRank of a page is divided evenly among the pages that it links to. Following equations are the calculations of plain and weighted PageRank of a page "A", respectively.

$$PR(A)=(1-d) + d*(PR(T_1)/C(T_1)+\ldots+ PR(T_n)/C(T_n))$$

$$PR(A)=(1-d) + d*([PR(T_1)*w(T_1)]/C(T_1)+\ldots+ [PR(Tn)*w(Tn)]/C(Tn))$$

In these calculations, page A is pointed by its parent pages $T_1, T_2, \ldots\ldots, T_n$. The parameter d is the dumping factor. In the weighted PageRank calculation, w(Tn) function refers to the weight given to a parent page.

In our implementation, we introduce six different top level domains and three different continents as choices of user preferences. These top level domains are commercial (.com), military (.mil), government (.gov), organization (.org), business (.net) and education. The continents are Asia, America and Europe. In order to calculate weighted PageRank values, we calculated $2^9 = 512$ different combinations of different personalized PageRank vectors. In calculating weighted PageRank, we find the weight correlations for all possible combinations of user preferences. We can illustrate our approach in finding the weight correlations with following example.

Example: ("edu", "ame") is one of the user preference combinations. As can be seen in Table-1, we favor the links that has "edu" top level domain or "ame" (america) region. If a Url happens to have both, then it gets the highest weight factor. In the end, we normalize the weights. Based on this example, a url http://www.indiana.edu would have the weight of "1", since it includes both top level domain "edu" and region "ame" (america).

edu	**1**	**x2**	**2**	**0.5**
ame	**1**	**x2**	**2**	**0.5**
gov	**1**	**x1**	**1**	**0.25**
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
edu_eur	**1**	**x2**	**2**	**0.5**
edu_ame	**1**	**x4**	**4**	**1**

Table – 1 Weight correlation table for combination ("edu", "ame")

To this end, we calculated 512 different weight correlations. When calculating the weighted PageRank of a page, we also calculated the weight correlation of a user profile preference combination and then retrieve the corresponding weight for each url.
In our implementation, we used compressed sparse row data structure for adjacency matrix to represent the web graph. We, also, used parallel arrays for vertices and personalized PageRank vectors. We implemented weight correlations as Java Objects. We used hash function to relate user preference combinations to their corresponding weight correlation objects.

PageRank calculation is heavily depended on the graph structure of the web. One important problem with PageRank calculation is the existence of danglink links (sinks) and source links. A danglink link is a page accumulating PageRank, however, never distributing the PageRank to other nodes. A source link is a page that never gets PageRanking from the rest of the graph.

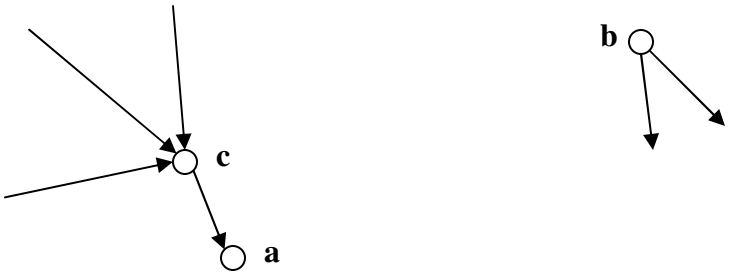


Figure – 1 Danglink and Source links

For example, consider two pages ‘c’ and ‘a’ where a does not reference to any other pages. During the iteration, ‘a’ will continually accumulate rank and never distribute the rank and therefore page ‘a’ will form a danglink link (sink). However, if we consider page ‘b’, during the iteration, page ‘b’ will always distribute the PageRank and never get any, therefore page ‘b’ will form a source.

We want PageRank values converge after certain iteration number. So, we introduce the notion of ‘magical node’ to connect sink links to source links. Magical node provides the means of distributing the ranks between the nodes of the web graph. We illustrated the magical node in Figure-2.

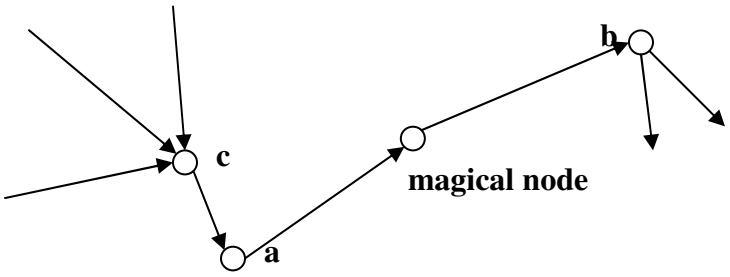
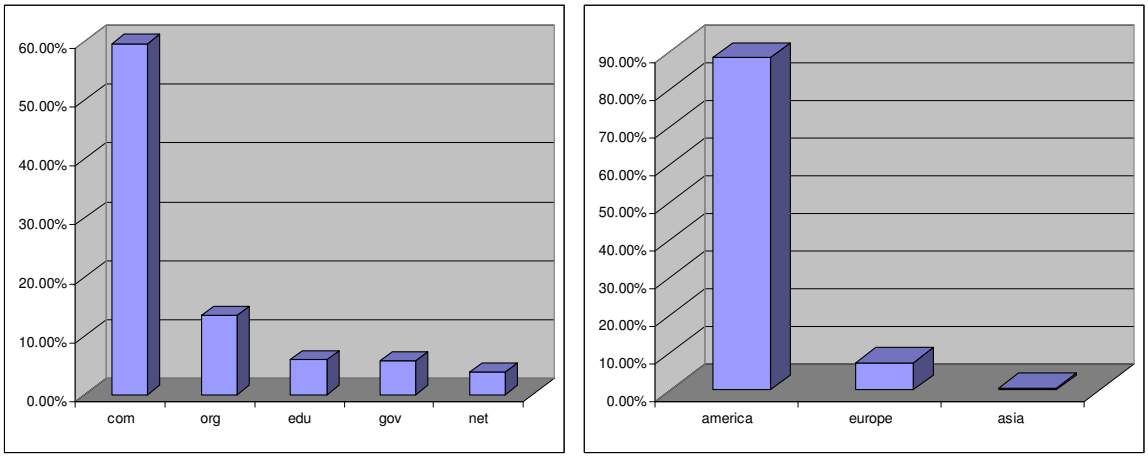


Figure – 2 Magical node to connect danglink and dource links

We experimentally found out that, such 'magical node' does not affect the final PageRank order. The main advantage of this is to allow us to include more nodes in our dataset, so that we can create a "good-size" connected web graph.

For the web graph, we crawled the web starting from some of the Yahoo sub-directories including "Education", "Regions" and "Government". As a result, we have a graph of 107890 Urls for our experiments. In order to avoid the disadvantages of sink and source nodes, we used the magical node idea. In total, including the edges that point to or from magical node, we have 468410 edges to connect the existing Urls in our graph. Unfortunately, the distribution of the top level domains and regions in our dataset did not turn out as we expected. The percentage of the commercial pages is dominant in our dataset. Likewise, our crawl data consists of mostly pages from continent America. Following graphs illustrate the distribution of top level domains and the regions in our dataset.



Graph -1 Top level domain and region distribution in our web graph

In the following sections of the paper we will be talking about our user interfaces and user studies in details.

User Interfaces

Our personalized web search engine designed as user friendly and easy to use application. The main goal is to reduce complications and involve users efficiently to contribute evaluations. Many of the research in this field that relies on user studies suffered from user biases. To overcome these difficulties, we aimed to provide a simple and efficient user interface.

User interface of the search engine has a modular structure and it is flexible for modifications. It consists of three parts. In the first part, users provide personal information that is optional (e.g. first and last name) and interests of domains that is

classified in two category. Top level domain is one category such as commercial, educational, organizational etc. domains, regional (continental) preferences divided by country extensions as illustrated in Figure-3. For example, .cn, .jp, .in extensions are counted as Asia. In each category users are able to select all options or some of them or nothing. Using these selections preference combination is calculated, and user queries ranked by associated PageRank matrix. In the next part, query page is displayed to the user and then user is asked to perform queries that expected to relevant to preference combination. The last step brings up top hits, as far as queries relevant user is free to submit them as shown in Figure-4. Also, users are able to browse the links in the next frame and navigate the pages. The whole process takes quite reasonable amount of time for users. However there are a lot of computations done in the background, user response time is still manageable.

In general, queries ranked in three different ways. First method assigns score values to query terms using TFIDF (Term Frequency – Inverse Document Frequency). Basically, score values measure relevance of query. Score values are embedded in Nutch index database [5]. The other two methods use PageRank matrixes and scores. At this step, the product of PageRank and score values is calculated and assigned to the hit. And then hit list resorted. The advantage of that is to favor higher score and PageRank values. The only difference in between plain and weighted PageRank is combination of matrixes. For example, plain PageRank matrix as in the first field, but weighted PageRank matrix is on 22^{nd} field (America and Education selected).

Collection of hits in three different sets are sorted and truncated at 10. Top 10 hits of each array is collected on single list by eliminating repeated hits. This new newly created collection is shuffled and displayed to the user so that to drop user biases at minimum. The user will not have any idea about the order of hit list. Then the user has to decide itself whether the links are relevant or in detail the content of the page that is browsed in the next frame is relevant. This approach ends up with a good intuition.

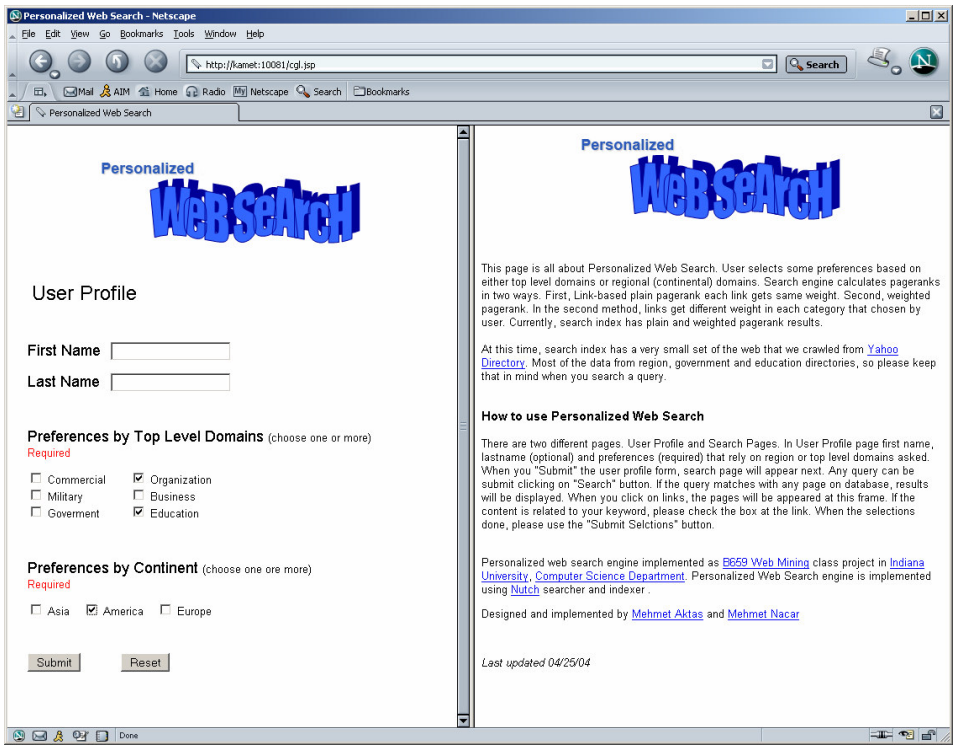


Figure-3: Introduction and user profile entry page

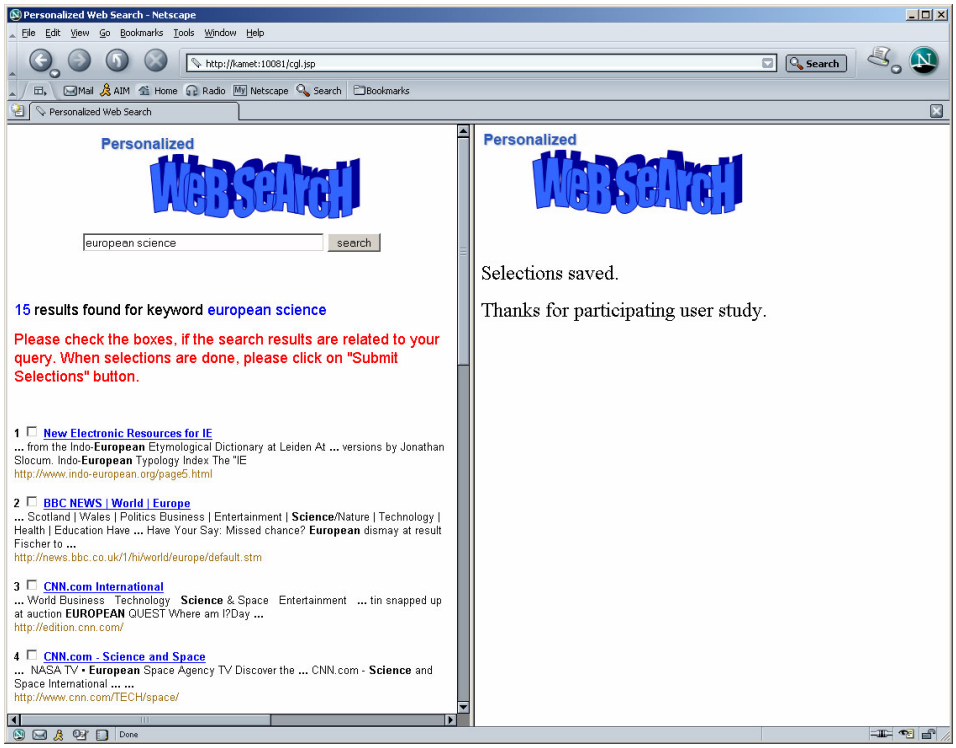


Figure-4: Query results and submission of user evaluations

Index Database and Searcher

Indexing and searching are important aspects of search engines that provide fast accesses to index database among queries. Search engines basically use neither relational databases nor flat files. But they use index based databases that makes look ups very fast so that queries are responded immediately. There are some works done in this area as open source project like Apache Lucene [6] and Nutch. In this project, we used nutch API and we add some adapter classes on Nutch.

Personalized search database contains recent crawl data from yahoo directory [7], called content.rdf file that collects urls and their topics as shown below . Nutch fetches all pages from the content file and indexes them accordingly. Fetching takes a while because all pages are loaded up to memory and then they are parsed into the terms.

```
<RDF xmlns:r="http://www.w3.org/TR/RDF/"
    xmlns:d="http://purl.org/dc/elements/1.0/"
    xmlns="http://directory.mozilla.org/rdf">

<Topic r:id="Top/Arts">
  <tag catid="2"/>
  <d:Title>Arts</d:Title>
  <link
r:resource="http://www3.bc.sympatico.ca/PHILLIPSHOTGLASS/GlassPage.html"/>
</Topic>

<ExternalPage
about="http://www3.bc.sympatico.ca/PHILLIPSHOTGLASS/GlassPage.html">
  <d:Title>John phillips Blown glass</d:Title>
  <d:Description>A small display of glass by John Phillips</d:Description>
</ExternalPage>

</RDF>
```

Terms are filtered eliminating stop words and stemming. Nutch uses PorterStemmer that implements Porter Algorithm for normalization of English words by stripping their extensions and is used to generalize the searches. For example, the Porter algorithm maps both 'search' and 'searching' (as well as 'searchnessing') to 'search' such that a query for 'search' will also match documents that contains the word 'searching' [6].

Index terms stored in the database based on vector space model. In other words, TFIDF based index system builds an inverted index with TF and IDF information. Nutch's built-in score function produces a quality-metric for similarity of queries in documents. Each hit would have a score value reflects the similarity of the document to the query term. In addition to score value, we also added a PageRank values to each documents. Because

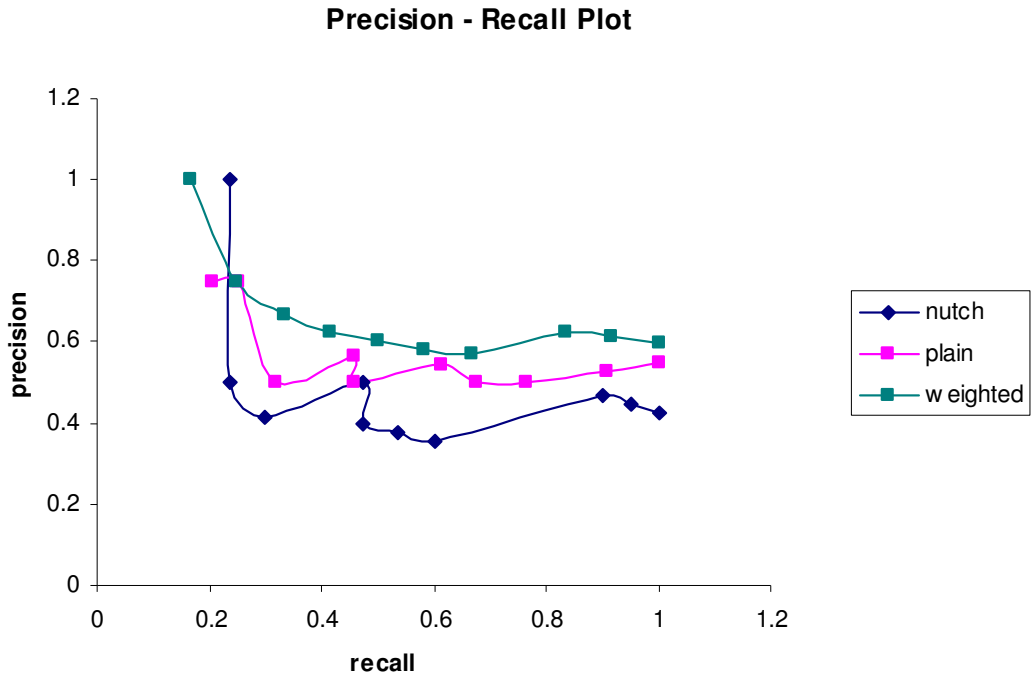
PageRanks are calculated using connected link structure of the web each hit also has PageRank value. Index database uses an extra field that is loaded with 512 different values of PageRank. Using these index scheme rather than flat files speeds up the application significantly.

Search operation is performed on the index which is a specialized data base that contains a pre compiled information of the document set. The index data base is optimized for locating quickly documents that contains certain words or terms. The index data base is created during the indexing process as explained before. The hit list is ordered by some measure of relevancy either ranking or scoring and may contain only a subset of the set of documents that matched the query (top 10 hits).

User Evaluations

User evaluation is a crucial criteria to find out whether weighted PageRanking improves the plain PageRanking. Feedbacks form users are keys to evaluation which is resulted with a conclusion. As long as user evaluations are important to get opinions about PageRanking methods, however user studies might lead biases too. To overcome that drawback we have used some simple manipulations on user interface as explained in great detail.

Precision-Recall graph is done on average of user feedbacks. Nutch score, plain PageRank and weighted PageRank precision-recall graphs showed below.



The graph shows that weighted PageRank gives better results. We managed to provide high precision at low recall. Whenever user gives right preferences, s/he will get more related results on top of the hit list. As can be seen on the graph, the fluctuations of the

other two method depends on number of user studies. Another reason could be user biases. For example, if user gets a few top hits and then s/he may tend to care not much about rest of the hits.

Conclusions

We discussed the details of our approach, which is to provide personalized web search by using weigted PageRank vectors. We discussed the evaluation method that we applied to compare three different ranking methods such as nutch, plain PageRank and personalized weighted PageRank ranking methods. Based on our results, personalized PageRank vectors improved the precision at low recall compared to other two ranking methods. However, we acknowledge the fact that our user studies are not based on a large dataset. Likewise, our connected web graph is too sparse and does not include a well distribution of top level domains and regions.

References

[1] ”The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Sergey Brin and Lawrence Page, http://www-db.stanford.edu/~backrub/ google.html

[2] “What can you do with a Web in your Pocket”
by Sergey Brin, Rajeev Motwani, Larry Page und Terry Winograd
http://www.research.microsoft.com/ research/db/debull/98june/webbase.ps

[3] “The PageRank Citation Ranking: Bringing Order to the Web”
by Lawrence Page, Sergey Brin, Rajeev Motwani und Terry Winograd
http://dbpubs.stanford.edu:8090/pub/ showDoc.Fulltext?lang=en&doc= 1999-66&format=pdf&compression=

[4 “Efficient Crawling Through URL Ordering”
by Junghoo Cho, Hector Garcia-Molina and Lawrence Page
http://dbpubs.stanford.edu:8090/pub/ showDoc.Fulltext?lang=en&doc= 1998-51&format=pdf&compression=

[5] “Nutch project”. www.nutch.org

[6] “ApacheLucene Project”. http://jakarta.apache.org/lucene/docs/index.html

[7] “Yahoo Directory”. www.yahoo.com

Appendix

A- A Collection of user data

Ana
Maguitman
ame+edu
22
mathematics
Nutch Score Hits
0 - http://education.yahoo.com/college/essentials/grad_search/grad_search.html --- 0.037956998
1 - http://www.acenet.edu/clll/ged/intro-A.cfm --- 0.030991761
2 - http://www.utexas.edu/cola/depts/lrc/numerals/numerals.html --- 0.030991761
3 - http://www.nsf.gov/od/lpa/news/publicat/start.htm --- 0.030991761
4 - http://www.acenet.edu/clll/ged/contacts.cfm --- 0.030991761
5 - http://www.jobs.irs.gov/mn-other3.html --- 0.030991761
6 - http://www.nsf.gov/od/lpa/priority/start.htm --- 0.030991761
7 - http://www.nsf.gov/od/lpa/sitemap.htm --- 0.030991761
8 - http://www.personal-loans.gb.net/ --- 0.021914486
9 - http://www.multcolib.org/homework/ --- 0.021914486
Plain PageRanking Hits
0 - http://www.gci275.com/peru/history.shtml --- [0.32044813164730135, 0.4372257607975305]
1 - http://education.yahoo.com/college/essentials/grad_search/grad_search.html --- [0.150405932562727, 0.15049111882838498]
2 - http://www.acenet.edu/clll/ged/intro-A.cfm --- [0.150405932562727, 0.15049111882838498]
3 - http://www.utexas.edu/cola/depts/lrc/numerals/numerals.html --- [0.150405932562727, 0.15049111882838498]
4 - http://www.nsf.gov/od/lpa/news/publicat/start.htm --- [0.150405932562727, 0.15049111882838498]
5 - http://www.acenet.edu/clll/ged/contacts.cfm --- [0.150405932562727, 0.15049111882838498]
6 - http://www.jobs.irs.gov/mn-other3.html --- [0.150405932562727, 0.15049111882838498]
7 - http://www.nsf.gov/od/lpa/priority/start.htm --- [0.150405932562727, 0.15049111882838498]
8 - http://www.nsf.gov/od/lpa/sitemap.htm --- [0.150405932562727, 0.15049111882838498]
9 - http://iume.tc.columbia.edu/academic.asp --- [0.21003839547787762, 0.21004181090507368]
Weighted PageRanking Hits
0 - http://www.gci275.com/peru/history.shtml --- [0.32044813164730135, 0.4372257607975305]
1 - http://education.yahoo.com/college/essentials/grad_search/grad_search.html --- [0.150405932562727, 0.15049111882838498]
2 - http://www.acenet.edu/clll/ged/intro-A.cfm --- [0.150405932562727, 0.15049111882838498]
3 - http://www.utexas.edu/cola/depts/lrc/numerals/numerals.html --- [0.150405932562727, 0.15049111882838498]
4 - http://www.nsf.gov/od/lpa/news/publicat/start.htm --- [0.150405932562727, 0.15049111882838498]
5 - http://www.acenet.edu/clll/ged/contacts.cfm --- [0.150405932562727, 0.15049111882838498]
6 - http://www.jobs.irs.gov/mn-other3.html --- [0.150405932562727, 0.15049111882838498]
7 - http://www.nsf.gov/od/lpa/priority/start.htm --- [0.150405932562727, 0.15049111882838498]
8 - http://www.nsf.gov/od/lpa/sitemap.htm --- [0.150405932562727, 0.15049111882838498]
9 - http://iume.tc.columbia.edu/academic.asp --- [0.21003839547787762, 0.21004181090507368]
User Selections
2 - http://www.utexas.edu/cola/depts/lrc/numerals/numerals.html --- [0.150405932562727, 0.15049111882838498]
0 - http://www.jobs.irs.gov/mn-other3.html --- [0.150405932562727, 0.15049111882838498]
8 - http://iume.tc.columbia.edu/academic.asp --- [0.21003839547787762, 0.21004181090507368]
7 - http://education.yahoo.com/college/essentials/grad_search/grad_search.html --- [0.150405932562727, 0.15049111882838498]
6 - http://www.nsf.gov/od/lpa/news/publicat/start.htm --- [0.150405932562727, 0.15049111882838498]

fulya
Erdinc
ame+edu+
22
video conferencing
Nutch Score Hits
0 - http://www.ols-english.co.uk/ --- 2.1412845
1 - http://www.paranoia.com/%7Edebaser/ --- 0.025792083
2 - http://www.ciscosolutioncenter.com/professionalsvcs/yahoo_portal_pro_services.html --- 0.025792083
3 - http://www.aarp.org/indexes/whatsnew.html --- 0.025792083
4 - http://www.apple.com/hardware/ --- 0.020374637
5 - http://www.pueblo.gsa.gov/cic_text/employ/employ-interview/emp.htm --- 0.015049821
6 - http://www.cisco.com/ --- 0.0150157055
7 - http://sdc.shockwave.com/go/gnav_home/ --- 0.011809459
8 - http://sdc.shockwave.com/go/gnav_sign_out/ --- 0.011809459
9 - http://www.macromedia.com/go/gnav_home/ --- 0.011809459

Plain PageRanking Hits
0 - http://www.ols-english.co.uk/ --- [0.150405932562727, 0.15049111882838498]
1 - http://www.aarp.org/indexes/whatsnew.html --- [0.17859899979155322, 0.27815355503811057]
2 - http://www.paranoia.com/%7Edebaser/ --- [0.150405932562727, 0.15049111882838498]
3 - http://www.ciscosolutioncenter.com/professionalsvcs/yahoo_portal_pro_services.html --- [0.150405932562727, 0.150491118828]
4 - http://www.apple.com/hardware/ --- [0.150405932562727, 0.15049111882838498]
5 - http://sdc.shockwave.com/go/gnav_home/ --- [0.21542290425857258, 0.21863680832737906]
6 - http://www.pueblo.gsa.gov/cic_text/employ/employ-interview/emp.htm --- [0.1530740427601493, 0.16960642547355387]
7 - http://www.cisco.com/ --- [0.150405932562727, 0.15049111882838498]
8 - http://www.macromedia.com/go/gnav_home/ --- [0.18461459423174553, 0.18637903452877244]
9 - http://www.macromedia.com/go/gnav_sign_out/ --- [0.1546889515504952, 0.1546889515504952]
Weighted PageRanking Hits
0 - http://www.ols-english.co.uk/ --- [0.150405932562727, 0.15049111882838498]
1 - http://www.aarp.org/indexes/whatsnew.html --- [0.17859899979155322, 0.27815355503811057]
2 - http://www.paranoia.com/%7Edebaser/ --- [0.150405932562727, 0.15049111882838498]
3 - http://www.ciscosolutioncenter.com/professionalsvcs/yahoo_portal_pro_services.html --- [0.150405932562727, 0.150491118828]
4 - http://www.apple.com/hardware/ --- [0.150405932562727, 0.15049111882838498]
5 - http://sdc.shockwave.com/go/gnav_home/ --- [0.21542290425857258, 0.21863680832737906]
6 - http://www.pueblo.gsa.gov/cic_text/employ/employ-interview/emp.htm --- [0.1530740427601493, 0.16960642547355387]
7 - http://www.cisco.com/ --- [0.150405932562727, 0.15049111882838498]
8 - http://www.macromedia.com/go/gnav_home/ --- [0.18461459423174553, 0.18637903452877244]
9 - http://www.macromedia.com/go/gnav_sign_out/ --- [0.1546889515504952, 0.1546889515504952]
User Selections
3 - http://www.cisco.com/ --- [0.150405932562727, 0.15049111882838498]
2 - http://sdc.shockwave.com/go/gnav_sign_out/ --- [0.150405932562727, 0.15049111882838498]
1 - http://www.macromedia.com/go/gnav_home/ --- [0.18461459423174553, 0.18637903452877244]
0 - http://sdc.shockwave.com/go/gnav_home/ --- [0.21542290425857258, 0.21863680832737906]
9 - http://www.ols-english.co.uk/ --- [0.150405932562727, 0.15049111882838498]
8 - http://www.ciscosolutioncenter.com/professionalsvcs/yahoo_portal_pro_services.html --- [0.150405932562727, 0.15049111882838498]
6 - http://www.macromedia.com/go/gnav_sign_out/ --- [0.1546889515504952, 0.1546889515504952]