BLOGRANK: Ranking on the blogosphere

Apostolos Kritikopoulos
Athens University of Economics and Business
Patision 76, Athens, Greece
+30 6977687978
apostolos@kritikopoulos.info

Martha Sideri
Athens University of Economics and Business
Patision 76, Athens, Greece
+30 2108203149
sideri@aueb.gr

Iraklis Varlamis
Athens University of Economics and Business
Patision 76, Athens, Greece
+30 2108203160
varlamis@aueb.gr

ABSTRACT
Although, the Blogosphere is part of the World Wide Web, weblogs present several features that differentiate them from traditional websites: the number of different editors, the multitude of topics, the connectivity among weblogs and bloggers, the update rate, and the importance of time in rating are some of them. Traditional search engines perform poorly on blogs since they do not cover these aspects. We propose an extension of Pagerank, which analyses and extends the link graph, in an attempt to exploit some of the weblog features. The analysis of the weblogs' link graph is based on the assumption that the visitor of a weblog tends to visit relevant or affiliated weblogs. Our algorithm, BlogRank, models the similarity between weblogs by incorporating information on common users, links and topics and generates a global ranking for all weblogs in a set. To validate our method we ran experiments on a weblog dataset, processed and adapted to our search engine:

http://spiderwave.aueb.gr/blogwave

Our experiments suggest that our algorithm enhances the quality of returned results.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information filtering, Relevance feedback

General Terms
Algorithms, Management

Keywords
Ranking, related blogs, Graph based ranking, PageRank

1.INTRODUCTION
'Blogging' is a popular way of publishing information on the web. Due to its ease of use, blogging has become a tool for web based communication, collaboration, knowledge sharing, reflection and debate 8. According to the weblog search engine of Technorati (http://technorati.com/weblog) that tracks 50 million weblogs, the size of blogosphere doubles every 6.5 months 8 and today is 100 times bigger than 3 years ago.

Every weblog consists of a series of entries, namely posts, which express uniform or contradictory opinions and link to other posts or web pages 8. Posts may contain text, images, videos and links, and depict personal beliefs. Links are used as suggestions, as a means to express agreement or disagreement 8 or even in order to bias search engine results (i.e. spam weblogs, Google bombs).

Apart from the explicit relation expressed by hyperlinks, weblogs can be related based on more implicit features, such as common topics, authors or interests. Two weblogs with posts on the same topics are relevant even when there is no explicit hyperlink between them. Two weblogs with shared authors or commentators are related implicitly. This relation is not necessarily supported by hyperlinks between posts.

Explicit hyperlinks between posts, and implicit relations between weblogs that emanate from the number of common bloggers or topics, turn the blogosphere into an interesting multi-colored and weighted graph. The graph's nodes correspond either to the weblogs or to their posts and the edges to hyperlinks or implicit links. The weights express the strength of relation between nodes (i.e. the number of hyperlinks, or the number of common authors or topics).

Although the graph formed by the hyperlinks between weblog posts is part of the web graph, the ranking algorithms for web pages (i.e. PageRank) seem to be insufficient for the following reasons:

- The number of links between weblog entries is very small. Thus, the weblog entries graph is very sparse and the ranking algorithms do not perform well.
- Weblog-specific information (time, topic, editor etc.) is not exploited in its full extent.

To improve the ranking results of PageRank on Weblogs we propose the following method:

- We first process the weblog graph from to provide a denser graph.
- We assign weights to the new edges taking into account several criteria, such as similarity in topics and

contributors between the source and target nodes, the number of explicit hyperlinks between nodes, and the difference in time of creation between the source and target node. The weights are assigned in a way that new hyperlinks are promoted.

- We modify in the standard PageRank algorithm to incorporate these criteria

We test the efficiency of our ranking method in a sample weblog dataset provided by Nielsen BuzzMetrics, Inc. For this reason we develop an experimental search engine over the dataset and allow web users to provide human judgment on the results. We use both implicit and explicit human evaluation measures in order to evaluate our algorithm.

The contributions of our work comprise: a blog ranking algorithm that exploits many of the weblog intrinsic features and reveals a new way for ranking weblogs, a blog search engine that can be extended to cover larger parts of the blogosphere, a metric of user satisfaction that exploits implicit 8 and explicit user feedback. With the aid of this metric we are able to measure the relevance of a page to the query and therefore evaluate the ranking algorithm.

The rest of this paper is structured as follows: Section 2 presents existing work in weblog analysis. Section 3 discusses the details of the dataset. In the same section we present our conception of the enhanced weblog graph and give examples on the different types of implicit links and on the final result. Section 4 presents in detail the ranking algorithm, describes how the enhanced weblog graph has been exploited and gives the motive behind our decisions. Section 5 illustrates our experiments with the dataset and the search engine. In separate subsections we present how we extended the graph, give the top ranked weblogs and discuss the setup and results of the user evaluation process. Section 6 contains our conclusions and presents the future plans of our work in this area.

2.RELATED WORK

The work presented in this paper is based on our previous experience in ranking nodes into highly connected graphs. The most significant output of our work in this direction is SpiderWave 8, a research search engine for the Greek portion of the Web (about 4 million documents, basically the .gr domain) designed by our research group. The engine can be reached from the Web site of our University (http://spiderwave.aueb.gr) and is an alternative search engine for Greek content. SpiderWave has an algorithm for ranking every page based only on the Greek fragment of the Web graph. For this paper we made available a test service for weblogs. The service can be accessed at:

http://spiderwave.aueb.gr/Blogwave

The majority of ranking algorithms that detect authority nodes in an interconnected graph (8, 8, 8) reside on the density of the graph. In the case of weblog entries, authors usually record their personal opinion on a topic, link to newspaper articles or websites, but rarely link to the opinion of other bloggers. This results to a very sparse weblog graph in which the use of a PageRank-like algorithm is problematic.

In order to bridge non-interconnected documents (i.e. non-hypertext documents), Kurland & Lee 8 assume additional links between documents with similar content. Moreover, they direct these links from the more to the less diverse documents. However, the analysis of content, a prerequisite in this approach, can prove very time consuming and inefficient for the quickly evolving blogosphere.

Fujimura et al 8 notice the problem of the weakly interconnected weblog graph and add a reputation property into every node in the graph in order to bias the random surfer model. The EigenRumor algorithm examines the graph of weblog posts and produces hub, authority and reputation scores to weblogs and authors and consequently to the posts.

The sparseness of the weblog graph has already been noticed by researchers and a number of implicit links have been created to increase the density of the graph. The implicit links suggested by Adar et al. 8 denote similarity between nodes in content and out-links. Although author name is not always characteristic on the author's identity (authors may have double identities inside a weblog, or the same username can be used in many weblogs by more than one user) if properly used can a supportive factor on the similarity between weblogs. For weblogs in the same server the username is unique for each registered user 8. The combination of weblog server name and user name can be consequently used as user id. As a result, the number of authors in common between two weblogs strengthens the implicit link between them.

The dataset we process contains links to non-weblog URLs (marked as Press). These are web pages' URLs and are ending nodes on our graph, since they have only incoming links. The ranking of such nodes is based on the number of incoming links and the authority of their referees. Such a ranking will show the most influential 8,8 non-weblog nodes for the weblogs community.

The result of running a ranking algorithm in the graph of weblog posts will be to find highly interconnected posts that have been referenced by many and refer to many posts. Although top ranked posts are very interesting for someone to read, it would be more interesting in an analysis of the blogosphere to find those weblogs that rank high in importance by collecting many influential posts. In order to achieve such a ranking we need to abstract the initial graph of posts to a graph of weblogs. We plan to do this weblog ranking in the future.

3.STRUCTURE

3.1The dataset

In order to explain our approach, we briefly present the structure of the specific dataset we used and the features we employed in our analysis. Although the weblog structure is not standard, all weblogs share the following structure: A weblog contains one or more posts and has a URL. Every post consists of an author, a body, a date and time of publishing and a URL of the full, individual article (the permalink). A post optionally includes: comments of readers, category tags and links to referrers (trackback links). Trackback and comment information is unavailable from the test dataset.

Topic information is available for 23.3% of the posts (for 3.6 million out of 15.4 millions weblog entries). The ***author*** name is not always very useful, mainly due to:

- anonymous posting. It is an option in several weblogs to allow users to post entries without providing a name.
- common user names. Several user names (i.e. admin, webmaster, moderator etc) appear in more than one weblogs but we can easily assume that correspond to different persons

- double identities. A person can have different usernames in different weblogs and less possible but probable to have two accounts in the same weblog.

Usually, members of a community contribute in more than one weblogs based on the interests they have in common. Even when there are no intrinsic links between two weblogs (hyperlinks, permalinks), an overlap between the contributor names indicates a relation. Bigger the overlap indicates stronger relation. As a result, we decide to employ author name information as a factor of relation between weblogs and we assign to the factor a weight of importance which can be changed upon case.

Similarly the choice of topic is subjective to the author. However, when combined with hyperlinks the analysis of topic and author information gives an important aspect of the blogosphere: authors that link to other authors, linked – related topics etc.

The ***date*** and ***time*** that an entry was registered is another useful piece of information. Analysis of entries based on date and time, will reveal more or less recent weblogs, more or less active weblogs and authors 8, and topics with short or long lifecycle. In this work, we exploit time information only when there is a hyperlink between posts of two weblogs. In this case, we take into account the difference in time between the two posts and aggregate this information (by taking the average difference in time of post) for all hyperlinked posts of the two weblogs. This is a way for promoting fresh hyperlinks. For example, when the post p_1 of weblog A links to the post p_2 of weblog B a few hours after p_2 is posted, and to the post p_3 of weblog C several days after p_3 is posted, the relation between A and B is stronger than between A and C.

3.2The post and weblog graph

This section illustrates the transition from the post to the weblog graph, through a small example. In the post graph (Figure 1), the links between posts are presented with black arrows. The grey lines connect the author with his/her posts and next to each post is the topic it refers to. The portion comprises 11 posts with only 3 hyperlinks between them. Obviously the posts graph is very sparse and existing ranking algorithms will not perform well.

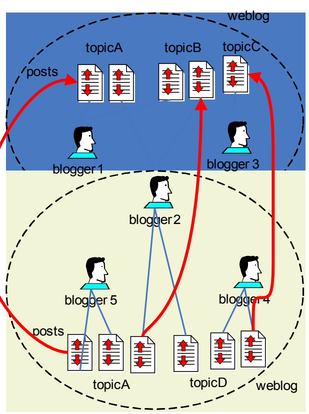

Figure 1. Hyperlinks and relations in post level

The weblog graph (Figure 2) aggregates information of the post graph. This produces a denser graph with two nodes connected with multiple edges. The nodes are weblogs comprising of a number of posts, authors and topics. The edges are: aggregated hyperlinks and similarity links based on similarity in authors and topics between the two blogs.

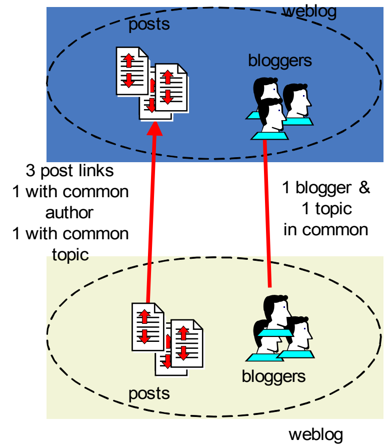

Figure 2. Links in weblog level

The proposed algorithm takes into account both link and similarity information in order to estimate the probability of a weblog surfer to follow a link to another weblog. The algorithm prioritizes links to weblogs with similar topics or contributors and assigns greater probability to weblogs which are strongly interconnected.

3.3Database structure

In order to process the dataset we used a relational database. The database schema has three entities: posts, tags and links. We have two sub-types of outlinks: links to posts and links to news URLs.

The following table and figure summarize the statistics of the dataset we used (using unique URLs).

Table 1. Statistics on the graph of posts

Nodes		Edges	
Crawled posts URLs	# news' URLs	Links to all posts	Links to news
15,456,270	172,454	1,430,222	334,854

The first two columns of Table 1 contain information on the nodes of our graph (permalinks with content, news URLs). The other two columns provide information on the edges of the graph (links to all permalinks, links to news URLs). By dividing the number of edges to the number of nodes we get approximately 0.093 links per post in the sparse post graph.

Table 2 gives useful statistics on the weblog graph.

Table 2. Statistics of the weblog graph

Nodes	Edges	
unique weblogs	Aggregated links between weblogs	Aggregated links from weblogs to news URLs
3,193,958	182,627	334,854

The second and third column, contain the number of unique links among weblogs and between weblogs and news' URLs. As expected the new graph has significantly fewer nodes (3.1 instead of 15.4 millions) and edges (0.18 instead of 1.4 million). As a result, the ranking algorithm will perform faster in the new graph. The algorithm will produce a global ranking of weblogs and not of individual posts. This ranking must be updated frequently in order to include every new post, weblog or link added to the graph. The ranking of weblogs is more stable, to the addition of a new post or a new link, than the ranking of posts, so the frequency of updates of the rank can be smaller.

In order to increase the number of edges in the graph we generate more implicit links based on similarity in authors and tags. We add a link between two weblogs when they have many authors in common or when their posts have many tags in common. After experimentation we decided on the minimum values for the number of authors and tags. This was the initial tuning of the parameters of our algorithm. We plan to fine tune the graph enhancement process in the future.

For the experiments we created a graph of weblogs. In order to exploit the information conveyed by the links to news URLs we employed the coupling factor 8 used in bibliography research as an indication of similarity between nodes. When two nodes (in our case weblogs) point to many news URLs in common we consider that the nodes are related.

The output of the algorithm, which we explain in the next section, is a ranking of the 3.1 millions weblogs in the dataset. However, the dataset does not contain posts from all these weblogs. So it is very interesting to crawl on the posts of all the weblogs.

4.The BLOGRANK algorithm

The output of our algorithm is a ranking of all weblogs in the dataset. This overall ranking is used by our search engine for the presentation of results: matching entries from highly ranked weblogs are presented first. In the case of entries from the same weblog, most recent entries are presented first. BlogRank is a generalized approach of Pagerank 8. The BlogRank of a Weblog A is given by the formula:

$$B(A) = (1\text{-}E) + E\,(FN(U_1 \rightarrow A) * B(U_1) + \ldots + FN(U_n \rightarrow A) * B(U_n)) \quad (1)$$

where: B(A) is the BlogRank of weblog A,

B(Ui) is the BlogRank of weblog Ui which link to weblog A,

E is a damping factor between 0 and 1 (normally is 0.85)

FN(Un→A) is the probability that a user who visits weblog n selects weblog A as a next visit, and denotes a factor which shows how much the weblog Un « fancies » weblog A.

The following equation holds:

$$\sum_{j=1}^{t} FN(Uz \rightarrow j) = 1 \quad (2)$$

where: z is a weblog with t outlinks (to other weblogs)

FN(Uz→j) is the possibility that the user will choose weblog j

If we assume in BlogRank that FN(Uz→j)=1/N where N is the total number of outlinks in weblog z, then we can easily derive the PageRank formula. We strongly believe that a user is not attracted equally by every outlink that exist in post of a given weblog. The most probable case is that the user was driven to a post because she was looking for topic or she is interested for the main subject of the post. It is logical to hypothesize that she is most probably going to continue her quest, by selecting similar post or news. From all the outlinks of weblog z, the significant function FN(Uz→j) favours those posts of the j weblog that:

a) belong to common categories with the weblog z

b) same users have posted as in weblog z

c) link to the same news posts as mentioned in weblog z

Before we apply the BlogRank, we expand the connected graph of the weblogs by adding bidirectional links between the weblogs that share same categories, users and news. Then we apply weights to every connection. The utility function that gives the possibility of user to move to weblog j once in z is :

$$FN(U_{z\rightarrow j}) = \frac{F_{z\rightarrow j}}{\sum (F_{z\rightarrow x})}$$

(3a)

Where

$$F_{z\rightarrow k} = L_{z\rightarrow k} + w_T * T_{z\rightarrow k} + w_A * A_{z\rightarrow k} + w_N * N_{z\rightarrow k} + w_D * D_{z\rightarrow k}$$

(3b)

And

L is the number of links from weblog z to weblog j

T is the number of common tags/categories between z and j

A is the number of authors that have posted in both z and j

N is the number of couplings of z and j to news URLs.

D equals to 24*60 / average(posting time difference in minutes), between z and j (only for hyperlinked posts)

$\mathbf{W_T}$, $\mathbf{W_A}$, $\mathbf{W_N}$, $\mathbf{W_D}$ are the weights we use in each one of the factors T,A,N,D respectively.

We generated 43 different rankings using different values for the parameters of formula 3b. We used human judgments to decide on the importance of the top-40 weblogs of every ranking. Experimentally we adjusted weights of formula 3b to the following values: W_T=1.70, W_A=1.10, W_N=4.80, W_D=0.40. The aim was to maximize user satisfaction from the top ranked results, and consequently from the results of every individual query. Although the selected weights are not fine-tuned, and an extended evaluation is under consideration, the first results we have available (see section 5.3) show that BlogRank outperforms the rest of the algorithms we tested.

In the following we give a short example on how we calculate Function 3b for the weblogs in Figure 2. The probability for a user that reads a post in the bottom weblog to follow a link to a post in the top weblog depends on the relevance between the two blogs. Function 3b takes into account the number of links between posts of the two blogs ($L_{bottom\rightarrow top}$=3), the number of common tags between posts of the two blogs ($T_{bottom\rightarrow top}$=1) and the number of common authors ($A_{bottom\rightarrow top}$ 1). The time and couplings information is ignored in this example. As a result, the nominator of FN($U_{bottom\rightarrow top}$) will be 5.8 (3+1.7*1+1.1*1=5.8). For the denominator we should consider the whole graph and take into account all the links between the bottom weblog and other weblogs.

5.EXPERIMENTS

5.1Extending the graph

By adding implicit links on the graph we increase its density and consequently the performance of the ranking algorithm. Of course, the notion of the web surfer, on which PageRank is based, is more obfuscated if we consider implicit links. In the extended graph, we consider the probability that the reader of a weblog A will move to a related weblog B, even when A and B are not hyperlinked. This transition requires additional steps (i.e. a recommendation engine that suggests "similar" posts), but is feasible with existing technologies 8.

We create implicit links by setting thresholds in the similarity factors. The thresholds have been set up experimentally and possibly need further tuning. Moreover, we adjust the four weights in formula 3 and test three combinations of weights.

The thresholds are as follows:

- The minimum required number of common tags is 3. Very rare tags (those that appeared in less than 500 weblogs) were excluded from the experiment. This results in 55,916,848 tag similarity links between weblogs.
- The minimum required number of common authors is 2. This results in 14,166,996 author similarity links. Authors with common username (such us "Admin", "John" or "Webmaster") were excluded.
- The adequate coupling factor is set to 2. This results in 4,235,634 news similarity links.

Since many of the implicit links between weblogs overlap the final extended graph contains a total of 74,384,925 distinct edges. We decided to use three different sets of weights (W_T, W_A, W_N) in our experiments., the following three cases:

We set the three weights to 0 and L=1. This means that we collapse the multiple links between two weblogs into a single link and ignore all implicit links. This gives the simple PageRank formula. We refer to this formula as Rank1 (PageRank).

We set the weights to 0 and make L equal to the number of distinct links between posts. We still ignore the implicit links but consider the multiplicity of links. This ranking is an extension of the PageRank algorithm and we call it Rank2 (or XRank). This ranking considers only one type of edges (hyperlinks) in the graph of weblogs but assigns weights based on the number of links between weblogs and is an average solution between PageRank and BlogRank.

Finally after a few experiments we decided on setting the weights to (2, 1, 3) for (common tags, common authors and common links to press articles) respectively, in an attempt to subjectively judge the importance of each type of implicit link. L still represents the cardinality of links. We call the formula Rank3 (BlogRank).

We notice that the weblogs that are ranked with Rank1 have similar order with the ones that were ranked with Rank2. But the order of Rank3 seems different from the previous two rankings.

Table 3. The 10 best ranked weblogs of Rank 1 (PageRank)

Weblog	Rank 1
http://www.msnbc.msn.com	344.35
http://www.webmink.net	246.76
http://www.boingboing.net	215.13
http://www.ehlc.net/news.html	185.54
http://www.bandelier.net	174.71
http://www.engadget.com	115.72
http://www.dailykos.com	112.94
http://thinkprogress.org	100.31
http://daoureport.salon.com	98.90
http://www.nuketown.com	95.67

Table 4. The 10 best ranked weblogs of Rank 2 (XRank)

Weblog	Rank 2
http://www.thecancerblog.com	180.01
http://ask.metafilter.com	140.03
http://www.engadget.com	118.07
http://www.tvsquad.com	103.25
http://www.greatestjournal.com	94.99
http://www.technudge.com	92.42
http://pajamasmedia.com	89.15
http://www.grassrootspa.com	74.77
http://www.luxist.com	71.22
http://www.highonserp.com	69.68

Table 5. The 10 best ranked weblogs of Rank 3 (BlogRank)

Weblog	Rank 3
http://www.bandelier.net	638.26
http://www.webmink.net	556.12
http://www.boingboing.net	301.98
http://www.engadget.com	281
http://thinkprogress.org	217.82
http://daoureport.salon.com	191.75
http://www.dailykos.com	185.94
http://www.techcrunch.com	151.11
http://www.ehlc.net/news.html	142.79
http://www.crooksandliars.com	137.68

In the results we notice that there are 147 common weblogs in the first 1000 ranked weblogs of each ranking type. The percentage (14.7%) denotes that the algorithms present different results and rank the weblogs in different order. In order to find the ranking algorithm that mostly satisfies users we setup a search engine for searching posts content (section 5.2) and have a group of users to make queries and evaluate the results without being aware of the ranking algorithm used in every case (section 5.3). Our claim is that the BlogRank outperforms the other rankings in terms of users' satisfaction. And the end of this section we present comparative results that prove our claims.

5.2Evaluation

5.2.1Ranking and User Satisfaction

The primary aim of a ranking algorithm, in the case of a web search, is to prioritize documents of higher importance to the user. A study in the related bibliography 8,8 shows that ranking algorithms are performed either on the full set of documents in the repository or in the subset of documents that match user's criteria. In the former case, the algorithms work offline, periodically examine the complete document set and produce a global ranking for all documents, irrespectively of the user's query. It is obvious that the latter case is faster since fewer documents are ranked. Moreover, these ranking algorithms can exploit the information carried by the search criteria (i.e. the query terms) and achieve a better ranking. However, since the ranking of query results is performed online, it should be very fast.

Of course, it is difficult to decide whether a ranking is better than another. There are mainly two approaches in evaluating the quality of a ranking algorithm. Both approaches consider that users submit queries and get lists of URLs as a reply. URL contents can be relevant to the query or not.

The first approach assumes that the relevance of the document to the query is known in advance (TREC tests, http://trec.nist.gov). As a result, the quality of the ranking algorithm is measured in top-K ranked documents, using metrics such as recall (number of relevant documents in the top-K divided by all the relevant documents in the set) and precision (number of relevant documents in the top-K divided by K).

The second approach assumes that there is no relevance information for the documents. In this case, the relevance of a result to the query is provided as a feedback by the users and represents user satisfaction. User feedback can be either implicit or explicit depending on the way user satisfaction is captured. Since we have no relevance information we are going to use this second approach in our experiments.

Approaches that use explicit feedback 8 ask the user to give a positive grade to each result and calculate the average grade for each set of results. Approaches that use implicit feedback assume that users are presented a ranked set of links to the documents accompanied by a title, a URL and a short description and based on this information they decide and click on the most relevant link.. The order of clicks, the time spent in each link, the number of clicked documents, the time spent in reading a description etc, are useful feedback information for evaluating the algorithm. Evaluation is performed without the user being aware of it.

Joachims 8 introduces techniques based entirely on clickthrough data to learn ranking functions. In this paper he makes experiments relating clickthrough data with the user's intentions. The FairPairs algorithm 8 attempts to avoid the presentation bias by switching the order in which top ranked results are presented. In our evaluation we compare three different rankings and we expect the user to rank the best results higher, so we do not avoid the bias. We believe that this is not a problem because the presentation bias is strong only if a well known engine (such as Google) is used in the experiments. Our implicit measure (Success Index, presented in the following section) takes into account the position of the clicked result as well as the order of the click in the clickstream. To our knowledge there is no implicit measure that includes all the results of Joachims 8 experiments. Our implicit measure agrees with the observation of Joachims that the user -more likely- is going to click only the top results of a query. Fox et al. 8 enrich clickthrough data with more behavioral data and developed Bayesian models to correlate implicit measures and explicit relevance. Authors in 8 identify a rich list of clickthrough features and defined a few evaluation metrics. However, for the evaluation of the ranking algorithm, explicit information was also used.

5.2.2Implicit and explicit evaluation

In our experiments we do not assume a priori knowledge neither on the ranking of documents nor on their relevance to every possible query. We rely on implicit and explicit users' judgments in order to define the quality of ranking.

Our primary aim is to evaluate user satisfaction for different ranking methods, using uninformed (blind) testing. The results presented to a user query, are ranked by one of the available ranking methods. The method was selected randomly each time. The evaluation was based on the posts selected and the order of selection. Since the users are not aware of the algorithm used for each query we are confident that the tests are totally unbiased. More details on the evaluation method are available at 8.

The typical use case for our search engine is that the user of BlogWave enters a query and chooses between the presented posts. The post is presented in a new window and the user is called to declare her satisfaction with a vote (a number between 1=not satisfied and 5=extremely satisfied). The user could vote many posts from the result set, although she can visit some posts without voting (we assume that the vote for these cases is 0).

The average user satisfaction (AUS) is the average of all votes:

$$AUS= \frac{\sum_{visited\_posts} vote_u}{|visited\_posts|} \quad (4)$$

In order to further enhance the evaluation process we use the Success Index (SI) metric which was presented in 8. The basic advantage of Success Index is that it does not require the user to vote for her satisfaction. BlogWave records the posts clicked on by the user, and the order in which they are clicked.

We then evaluate the user's response using Success Index, a number between 0 and 1:

$$SI = \frac{1}{n}\sum_{t=1}^{n} \frac{n-t+1}{d_t * n} \quad (5)$$

where: **n** is the total number of the posts selected by the user
dt is the order in the list of the **t**-th post selected by the user

The SI score rewards the clicking of high items early on. The reverse ranks of the items clicked are weight-averaged, with weights decreasing linearly from 1 down to 1/n with each click. For example, suppose n = 2 and the posts ranked 2 and 10 were clicked. If 2 is clicked first, then the SI score is bigger (27.5%); if it is clicked second, the SI is smaller (17.5%). More controversially, SI penalizes many clicks; for example, the clicking order 2-1-3 has higher score than 1-2-3-4. In the absence of rating (when the user visits the post but does not provide a score) we assign zero score to the post. However, in our experiments we excluded the queries for which we have no user feedback.

During our experiment period, we specifically asked many users to enter and use the BlogWave service in order to have as many evaluation data as possible in the short time given. In an attempt to eliminate subjective bias, the experiment was double-blind since neither the individual users nor we know in advance the ranking method that is used in every query (the method was selected randomly). The use of double-blind test, allows the comparison of ranking methods against different query sets, which is the case in our experiment and generally in web search engines. This information is stored in the database and is used only for the evaluation of user satisfaction.

5.3Setup

The hardware we used was only a simple server PC (Pentium 4, 3.2 GHz, 2Gb memory, 117 Gb and 378Gb hard drives, Microsoft Windows 2003 Server). At the same PC we hosted the search engine, the database with the corpus of Nielsen BuzzMetrics and the user information. For the experiments we extended the existing search engine of SpiderWave with the BlogWave service (figure 5) for searching the blogs in the dataset. The data was split in two SQL databases: one with the content (63 GB) and the second (31 GB) with all the entities such as post, tag, outlink and the necessary user information. The Lucene.Net (http://www.dotlucene.net) open source text search service was used for the indexing of the content, which needed 13 GB to populate the indexes. The response time for each query was between 1 and 15 seconds, depending on the complexity of the query (how common a query term was), the resources usage of the server, and the cached information that the text search service kept.

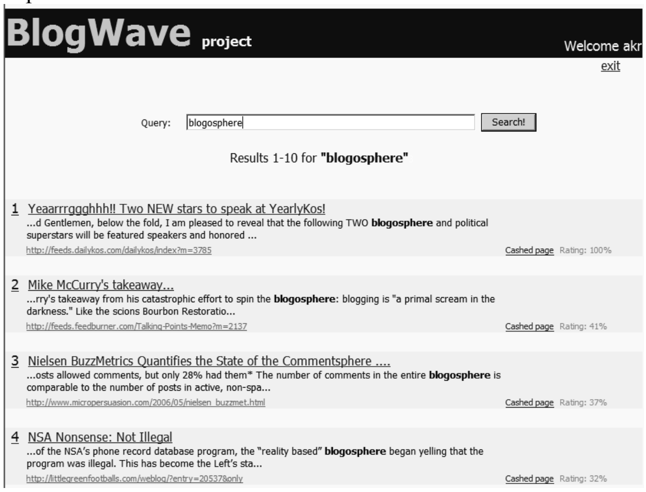

Figure 5. Evaluation of the search engine results

The result is a search engine on the content of blogs 8 which presents the results of a search to the user. The results are ranked using one out of three different sets of weights in the ranking formula (Function 3a). For every query the BlogWave executed a database query: at the beginning the 10000 most important posts were asked from the text search service. After that, the posts were ordered by importance of their weblog, depending on the ranking method used for the query. In case of a tie, then the post date is used to order the posts (most recent are presented first). Users evaluated the results by selecting the ranked posts.

The results we present to the user comprise posts ordered by the ranking of the weblog that they belong. In the case of two posts from the same weblog the most recent post gets a higher rank. Finally we grouped the query results, and computed the average SI score for each one of the three rankings we tested. Below we present the results of our experiment:

General

Time period of the Experiment: November–December 2006 (the experiment is still running, so we continuously collect user evaluation and we are able to tune the ranking formula)

Number of logged-in users: 102

Number of queries asked: 253

Number of queries ranked by the users: 84

Table 6. Average SI scores

Group A) Queries which were based on the Rank 1 - PageRank	
Number of queries:	78 (39 of them were ranked)
Average Success Index:	0.139
Average user satisfaction:	2.01 (in a 1-worst to 5-best range)
Group B) Queries which were based on the Rank 2 – XRank	
Number of queries:	87 (22 of them were ranked)
Average Success Index:	0.348
Average user satisfaction:	2.68 (in a 1-worst to 5-best range)
Group C) Queries which were based on the Rank 3-BlogRank	
Number of queries:	88 (27 of them were ranked)
Average Success Index:	**0.576**
Average user satisfaction:	**3.71** (in a 1-worst to 5-best range)

For group A the coin flipped and determined that the result set of posts will be sorted using the Rank 1 of each weblog that the post was belong to. For group B the engine sorted the presented queries by using the extended PageRank algorithm (Rank2). And finally for group C the results were sorted by using the BlogRank (Rank3). As we can see the average SI scores (0.576) in BlogRank's queries is much higher than the PageRank's (0.138) and XRank's (0.348) equivalent. The same happens with the AUS score.

Submitting the results of Table 6 to the t-Test statistical analysis we result that the observed difference between the means is significant, supporting the conclusion that the results of group C are substantial better that the results of the group B and that the results of group B are better than the results of group A. We can safely conclude that the best method is the Rank 3 - BlogRank which appears to considerably improve the quality of the retrieved information.

6.CONCLUSIONS – FUTURE WORK

We have proposed a method for using link graph characteristics, time and common attributes between the posts to enhance the quality of the results of the ranking mechanism for each weblog's importance. Our experimental results are quite encouraging. Of course, more experimental evaluation of our method, as well as tuning of its parameters is needed.

We developed and tested our method in the context of a very modest fragment of the Weblogging ecosystem. This scaled-down experimentation and prototyping may be an interesting methodology for quickly testing information retrieval ideas, and for expanding the realm of research groups, especially academic groups lacking strong industrial contacts, that are in a position to conduct search engine research. But does our method scale to the entire Blogosphere? First let's check if we could calculate the BlogRank for each weblog for the web. In our case (about 3.1 million weblogs) PageRank took 18 hours to complete, while BlogRank needed 21 hours (17% more time), which is a small difference. We therefore believe that out method can be scaled to the entire blogosphere, since it needs almost the same time as Pagerank.

Our plans for future research include the following:

We plan to explore more ways to use time information in the ranking of weblogs

We will focus on the use of objective information for describing the topics of posts and weblogs. For the topic detection process, instead of processing the Content and Tag information of a post, which is controlled by the author, we will use incoming hyperlinks information which is more objective 8.

We plan to process other aspects of the posts graph, more specifically, instead of grouping posts by weblog we plan to group posts "by topic" and "by author", thus forming a graph of interconnected topics and a graph of interconnected authors. The strength of each connection will be based on the number of real links between posts of each topic or author. Both author and topic graphs are directed, strongly connected and have many nodes. Using the biased surfer model we can estimate the probability of a surfer to follow a link to another topic or to another author's post thus revealing the most authoritative authors or topics 8.

Finally we will try to develop our implicit measure of user satisfaction to agree with all experimental observations.

7.ACKNOWLEDGMENTS

Our thanks to our students, who helped us with the experiments. To Nielsen Buzmetrics for the weblog dataset.

8.REFERENCES

[1] Adar, E., Zhang, L., Adamic, L., Lukose, R., (2004). "Implicit Structure and the Dynamics of Blogspace", Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, WWW 2004.

[2] Agichtein, E., Brill, E. and Dumais, S. T. (2006). Improving Web Search Ranking by Incorporating User Behavior Information, in the Proceedings of SIGIR.

[3] Efimova, L., de Moor, A. (2005). Beyond Personal Webpublishing: An Exploratory Study of Conversational Blogging Practices, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05.

[4] Eirinaki, M., Vazirgiannis, M., Varlamis, I. (2003). SEWeP: using site semantics and a taxonomy to enhance the Web personalization process. 9th ACM SIGKDD conference on Knowledge discovery and data mining, Washington, D.C.

[5] Fox, S., Karnawat, K., Mydland, M., Dumais S. T. , White. T., (2005). Evaluating implicit measures to improve the search experience. In ACM Transactions on Information Systems

[6] Fujimura, K., Inoue, T., Sugisaki, M., (2005). "The EigenRumor Algorithm for Ranking Weblogs", 2nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, WWW 2005.

[7] Gill, K., (2004), "How can we measure the influence of the blogosphere?", Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, WWW 2004.

[8] Joachims, T. , Granka, L. , Pan, B., Hembrooke, H, Gay, G., (2005). Accurately interpreting clickthrough data as implicit feedback, Proceedings of the 28th annual international ACM SIGIR conference.

[9] Radlinski, F., Joachims, T. (2006) Minimally Invasive Randomization for Collecting Unbiased Preferences from Clickthrough Logs, Proceedings of the 21st National Conference on Artificial Intelligence (AAAI).

[10] Kessler, M.M. (1963). "Bibliographic Coupling between Scientific Papers," Am. Documentation, vol. 14, no. 1.

[11] Kritikopoulos, A., Sideri, M. & Stroggilos, K. (2004). CrawlWave: A Distributed Crawler, 3rd Hellenic Conference on Artificial Intelligence.

[12] Kritikopoulos, A., Sideri, M. (2005). The Compass Filter: Search Engine Result Personalization Using Web Communities, Lecture Notes in Computer Science, Springer, Volume 3169, pp. 229 – 240

[13] Kumar, R., Novak, J., Tomkins, A. (2006). Structure and evolution of online social networks. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining

[14] Kurland, O., Lee, L. (2005). PageRank without hyperlinks: Structural re-ranking using links induced by language models, Proceedings of SIGIR, pp. 306-313.

[15] Langville, A., Meyer, C. (2003). Deeper Inside PageRank, Internet Mathematics Vol. 1, No. 3: 335-380, A K Peters Ltd.

[16] Massa, P., Hayes, C. (2005). Page-reRank: using trusted links to re-rank authority Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)

[17] Mishne, G., Glance N. (2006). Predicting Movie Sales from Blogger Sentiment. In proceedings of AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW 2006), 2006.

[18] Mishne, G., de Rijke, M. A Study of Blog Search, In: 28th European Conference on Information Retrieval (ECIR'06), 2006.

[19] Nakajima, S., Tatemura, J., Hino, Y., Hara, Y., Tanaka, K., (2005), "Discovering Important Bloggers based on Analyzing Weblog Threads", 2nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, WWW 2005.

[20] Oztekin U., Karypis, G., Kumar, V., (2003) Average Positive of User Clicks as an Automated and Not-Intrusive Way of Evaluating Ranking Methods, Univ. of Minnesota - Computer Science and Engineering. TR 03-009.

[21] Page, L., Brin, S., Motwani, R. & Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford, USA.

[22] Sudweeks, F., Simoff, S.J. (2005). Leading conversations: Communication behaviours of emergent leaders in virtual teams, Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38). Los Alamitos: IEEE Press.

[23] Sifry, D. (2006). "State of the Blogosphere, February 2006 Part 1: On Blogosphere Growth", weblog post. At: http://www.sifry.com/alerts/archives/000436.html

[24] Varlamis, I., Vazirgiannis, M., Halkidi, M., Nguyen, B., (2004). "THESUS, a Closer View on Web Content Management Enhanced with Link Semantics," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 6, pp. 685-700.Sannella, M. J. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Ph.D. Thesis, University of Washington, Seattle, WA, 1994.