CLBCRA-Approach for Combination of Content-Based and Link-Based Ranking in Web Search

Hao-ming Wang and Ye Guo

Department of Computer Science, Xi'an University of Finance & Economics,
Xi'an, Shaanxi 710061, P.R. China
{hmwang,yeguo}@mail.xaufe.edu.cn

Abstract. There were two kinds of methods in information retrieval, based on content and based on hyper-link. The quantity of computation in systems based on content was very large and the precision in systems based on hyper-link only was not ideal. It was necessary to develop a technique combining the advantages of two systems. In this paper, we drew up a framework by using the two methods. We set up the transition probability matrix, which composed of link information and the relevant value of pages with the given query. The relevant value was denoted by `TFIDF`. We got the CLBCRA by solving the equation with the coefficient of transition probability matrix. Experimental results showed that more pages, which were important both in content and hyper-link, were selected.

1 Introduction

With information proliferation on the web as well as popularity of Internet, how to locate related information as well as providing accordingly information interpretation has created big challenges for research in the fields of data engineering, IR as well as data mining due to features of Web (huge volume, heterogeneous, dynamic and semi-structured etc.). [1]

While web search engine can retrieve information on the Web for a specific topic, users have to step a long ordered list in order to locate the valuable information, which is often tedious and less efficient due to various reasons like huge volume of information.

The search engines are based on one of the two methods, the content of the pages and the link structure. The first kind of search engineers works well for traditional documents, but the performance drops significant when applied to the web pages. The main reason is that there is too much irrelevant information contained in a web page. The second one takes the hyperlink structures of web pages into account in order to improve the performance. The examples are Pagerank and HITS. They are applied to Google and the CLEVER project respectively.

R. Alhajj et al. (Eds.): ADMA 2007, LNAI 4632, pp. 23–34, 2007.

However, these algorithms have shortcomings in that (1) the weight for a web page is merely defined; and (2) the relativity of contents among hyperlinks of web pages are not considered. [2]

In this paper, we combine the contents and the links among the pages in order to refine the retrieval results. Firstly, we compute the the similarity of pages to the query, the TFIDF is often used. And then, we compute the new pagerank by the similarity and the out-link information of each page. As the page set, which is computed in this algorithm, includes all the pages we can find. The new pagerank is called `Content and Link Based Complete Ranking Algorithm(CLBCRA)`.

This paper is organized as follows: Section 2 introduces the concept of `Pagerank` and `TFIDF`. Section 3 describes the algorithm of `CLBCRA`. Section 4 presents the experimental results for evaluating our proposed methods. Finally, we conclude the paper with a summary and directions for future work in Section 5.

2 Related Works

2.1 Pagerank

PageRank was developed at Stanford University by Larry Page and Sergey Brin as part of a research project about a new kind of search engine. The project started in 1995 and led to a functional prototype, named Google, in 1998. [3,4]. The algorithm can be described as:

Let u be the web page. Then let F_u be the set of pages u points to and B_u be the set of pages that point to u. Let $N_u = |F_u|$ be the number of links from u and let c be a factor used for normalization (so that the total rank of all web pages is constant).

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}.$$

PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for any-size collection of documents.

The formula uses a model of a random surfer who gets bored after several clicks and switches to a random page. The PageRank value of a page reflects the chance that the random surfer will land on that page by clicking on a link. It can be understood as a `Markov chain` in which the states are pages, and the transitions are all equally probable and edges are the links between pages.

The PageRank values are the entries of the `dominant eigenvector` of the `modified adjacency matrix`. As a result of Markov theory, it can be shown that the PageRank of a page is the probability of being at that page after lots of clicks.

2.2 HITS

The HITS is another algorithm for rating, ranking web pages. [5] HITS uses two values for each page, the `authority` value and the `hub` value. Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Relevance of the linked pages is also considered in some implementations.

The algorithm is similar to PageRank, in that it is an iterative algorithm based purely on the linkage of the pages on the web. However it does have some major differences from Pagerank:

- It is executed at query time, and not at indexing time, with the associated hit on performance that accompanies query-time processing.
- It is not commonly used by search engines.
- It computes two scores per page (hub and authority) as opposed to a single score.
- It is processed on a small subset of relevant pages, not all pages as was the case with PageRank.

2.3 TFIDF

TFIDF is the most common weighting method used to describe documents in the Vector Space Model (VSM), particularly in IR problems. Regarding text categorization, this weighting function has been particularly related to two important machine learning methods: kNN (k-nearest neighbor) and SVM(Support Vector Machine). The TFIDF function weights each vector component (each of them relating to a word of the vocabulary) of each document on the following basis.

Assuming the document d is represented by vector $\tilde{d} = (\widetilde{w^{(1)}}, \widetilde{w^{(2)}}, ..., \widetilde{w^{(N)}})$ in a vector space. Each dimension $\widetilde{w^{(i)}}(i \in [1, N])$ of $\tilde{d}$ represents the weight of the feature w_i, which is the word selected from the document d. [6,7,8] The N means number of all features.

The values of the vector elements $w^{(i)}(i \in [1, N])$ are calculated as a combination of the statistics $TF(w_i, d)$(Term Frequency) and $DF(w_i)$ (Document Frequency).

$$w^{(i)} = TF(w_i, d) \times IDF(w_i).$$

Where $TF(w_i, d)$ is the number of word w_i occurred in document d. $IDF(w_i)$ can be computed by the number of documents N_{all} and the number of documents $DF(w_i)$, in which the word w_i occurred at least once time.

$$IDF(w_i) = log\frac{N_{all}}{DF(w_i)}.$$

We can construct the vector $\tilde{q}$ of a query q by using the similar way just as we do for the documents. The cosine distance between the vector $\tilde{d}$ and the $\tilde{q}$, which means the similarity of them, can be computed. The bigger the value, the more similar they are.

2.4 Computing the Pagerank

According to the Markov theory, the Pagerank is the `dominant eigenvector` of the `adjacency matrix`. In order to get the real, positive, and the biggest eigenvalue, the adjacency matrix should be changed in 2 steps:

- guarantees the matrix is row-stochastic;
- guarantees the matrix is irreducible by adding the link pair to each page.

The second step means add the links between all pages. In Ref. [9, 10], it was pointed out that the modification of matrix might change the eigenvector order of the matrix or change the importance of the pages. The author then went on to explain a new algorithm with the same complexity of the original PageRank algorithm that solves this problem.

2.5 Precision and Recall

For a retrieval system, there are 2 sides should be considered, the *precision* and the *recall*. Just as the illustrator in Fig.2.5.

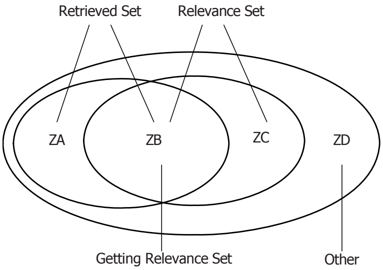


Fig. 1. Concept of Information Retrieval

For a given query Q, we can define,

- $ZA \cup ZB$: all retrieval pages set;
- $ZB \cup ZC$: all relevance pages set;
- ZA: pages set which retrieved but not relevance to the query;
- ZB: pages set which retrieved and relevance to the query indeed;
- ZC: pages set which relevance but could be retrieved;
- ZD: all the other pages set;
- Precision: $\dfrac{ZB}{ZA + ZB}$;
- Recall: $\dfrac{ZB}{ZB + ZC}$.

3 New Model of CLBCRA

We donate the query from the user with Q, all of the pages the retrieval system can select form the $Set\_1 = \{S_i, i \in [1, m]\}$. The pages in Set_1 link to other pages, which part of them belong to the Set_1, and the others form the $Set\_2 = \{S_i, i \in [1, n]\}$. All other pages in the system form the Set_3. [11,12,13]They are shown in Fig.2.

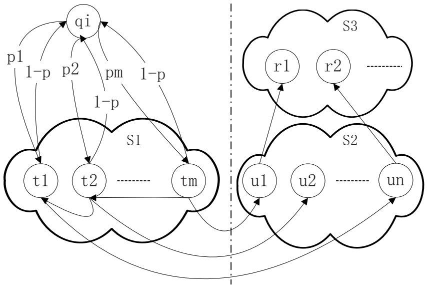


Fig. 2. New Model

In this model, there are 4 parts,

(1) Set S_0: all of the queries,which the user submits to the retrieval system, form the query set. In our illustrator, the $q_i \in S_0$;
(2) Set S_1: the pages are retrieved by the system according to the query q_i; just as the $t_1, t_2, \cdots, t_m$;
(3) Set S_2: the pages can be reached in 1 step from the pages in set S_1, just as the $u_1, u_2, \cdots, u_n$;
(4) Set S_3: all other pages out of the S_0, S_1 and S_2.

3.1 Transition Probability

When user submits a query to the retrieval system, he gets feedback results, S_1, from the system. Some of pages in S_1 are indeed relevant to the query and others are not. Meanwhile, some of pages in S_2 and S_3 are relevant to the query, but they are not retrieved by the system.

When user arrive the page in S_1, he will check the page carefully. If the page is relevant to the query, he will look through the page and move to other pages following the links. If the page is not relevant to the query, he has two choice: going back to the query or moving to other page randomly. In our experiment, we assume the user go back to the query.

(1) $q \rightarrow t_i, t_i \in Set\_1$

Assuming he moves from the q to the page $t_i, t_i \in S_1$ with the probability p_i, it is always true that $\sum_i p_i = 1$.

$$p_i = t_{0i} = \frac{\delta_{0i}}{\sum_i \delta_{0i}}. \tag{1}$$

where δ_{0i} is the relevant value of query q and page t_i. The value can be computed by many ways, such as the `TFIDF`. The value should be nonnegative.

(2) Page $i \rightarrow j$

No matter page i is in S_1 or not, the user should check it in order to determine whether the page relevant to the query q or not. We get,

$$\varSigma t_{ij} = \begin{cases} \mu & p_i(q) > 0 \quad \wedge \quad relevance(i,q) > 0; \\ 1-\mu & p_i(q) > 0 \quad \wedge \quad relevance(i,q) = 0; \\ \nu & p_i(q) = 0 \quad \wedge \quad relevance(i,q) > 0; \\ 1-\nu & p_i(q) = 0 \quad \wedge \quad relevance(i,q) = 0. \end{cases} \tag{2}$$

Where μ is the sum of probability the user jumps to other pages following the hyper-link from page i when page i is in S_1 and indeed relevant to the query q. Otherwise, the user jumps back to the q with the probability $1-\mu$.

The ν is the sum of probability the user jumps to other pages following the hyper-link from page i when page i is not in S_1 but it is indeed relevant to the query q. Otherwise, the user jumps back to the q with the probability $1-\nu$.

According to the Fig.2, we can define the transition probability T as,

(1) $(t_{0i}, \forall i)$
It means the probability from query q to page $i, i \in S_1$.

$$t_{0i} = \frac{\delta_{0i}}{\sum\limits_i \delta_{0i}}. \tag{3}$$

(2) $(t_{ij}, \forall i, j)$
It means the probability from page i, which is relevant to the query, move to page j. We assume that the out-links of page i have the same probability to be selected.

$$t_{ij} = \begin{cases} \mu * m_{ij} & p_i(q) > 0; \\ \nu * m_{ij} & Otherwise. \end{cases} \tag{4}$$

where $m_{ij} = \dfrac{1}{\sum\limits_j linknum(i \rightarrow j)}$.

(3) $(t_{i0}, \forall i)$
It means the probability of returning to query q when the user finds the page i is not relevant to the query q.

$$t_{i0} = \begin{cases} 1-\mu & p_i(q) > 0; \\ 1-\nu & Otherwise. \end{cases} \tag{5}$$

Gathering the q,S_1,S_2 and S_3, we get the transition probability matrix T shown in (6) .

$$\mathbf{T} = \begin{pmatrix} 0 & P'(q) & 0 & 0 \\ (1-\mu)U_1 & \mu * M_{11} & \mu * M_{12} & 0 \\ (1-\nu)U_2 & \nu * M_{21} & \nu * M_{22} & \nu * M_{23} \\ (1-\nu)U_3 & \nu * M_{31} & \nu * M_{32} & \nu * M_{33} \end{pmatrix} \tag{6}$$

where $P'(q) = (p_1, p_2, \cdots, p_n)$ is probability vector of query q links to all pages. $U_i, (i = 1, 2, 3)$ is the $n_i \times 1$ vector. And $n_i, (i = 1, 2, 3)$ is the page number of set $S_i, (i = 1, 2, 3)$. $M_{ij}, (i, j = 1, 2, 3)$ is the adjacency matrix of set $S_i, (i = 1, 2, 3)$.

In the formula (6), if the $\nu \ll 1$ is true, it can be changed to

$$\mathbf{T} = \begin{pmatrix} 0 & P'(q) & 0 & 0 \\ (1-\mu)U_1 & \mu * M_{11} & \mu * M_{12} & 0 \\ U_2 & 0 & 0 & 0 \\ U_3 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{T}' = \begin{pmatrix} 0 & (1-\mu)U_1 & U_2 & U_3 \\ P(q) & \mu * M'_{11} & 0 & 0 \\ 0 & \mu * M'_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{7}$$

3.2 Computing the Eigenvalue

Assuming $QQ = (x_0, X'_1, X'_2, X'_3)'$, we get $T' * QQ = QQ$.

$$\begin{pmatrix} 0 & (1-\mu)U_1 & U_2 & U_3 \\ P(q) & \mu * M'_{11} & 0 & 0 \\ 0 & \mu * M'_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} x_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

$$(1-\mu) * |X_1| + |X_2| + |X_3| = x_0; \tag{8}$$

$$x_0 * P(q) + \mu * M'_{11} * X_1 = X_1; \tag{9}$$

$$\mu * M'_{12} * X_1 = X_2; \tag{10}$$

$$X_3 = 0; \tag{11}$$

From (9),

$$\begin{aligned} (I - \mu * M'_{11}) * X_1 &= x_0 * P(q) \\ \Rightarrow X_1 &= x_0 * (I - \mu * M'_{11})^{-1} * P(q) \\ \Rightarrow X_1 &= x_0 * V. \end{aligned}$$

where $V = (I - \mu * M'_{11})^{-1} * P(q)$.

From (10),

$$X_2 = x_0 * \mu * M'_{12} * V.$$

From (8),

$$\begin{aligned} (1-\mu) * |x_0 * V| + |x_0 * \mu * M'_{12} * V| + |0| &= x_0 \\ \Rightarrow (1-\mu) * |V| + \mu * |M'_{12} * V| &= 1 \\ \Rightarrow (1-\mu) * |V| &= 1 - \mu * |M'_{12} * V| \\ \Rightarrow \mu * |M'_{12} * V| &= 1 - (1-\mu) * |V|. \end{aligned} \tag{12}$$

As $(x_0 + |X_1| + |X_2| + |X_3| = 1)$ is always true, we get

$$\begin{aligned} x_0 + |x_0 * V| + |x_0 * \mu * M'_{12} * V| + |0| &= 1 \\ \Rightarrow x_0 * (1 + |V| + |\mu * M'_{12} * V|) &= 1 \end{aligned}$$

Changing (12),

$$\begin{aligned} x_0 * (1 + |V| + 1 - (1 - \mu) * |V|) &= 1 \\ \Rightarrow x_0 * (2 + |V| - |V| + \mu * |V|) &= 1 \\ \Rightarrow x_0 * (2 + \mu * |V|) &= 1 \\ \Rightarrow x_0 &= (2 + \mu * |V|)^{-1}. \end{aligned}$$

Above all, we get

$$\begin{cases} x_0 = (2 + \mu * |V|)^{-1} \\ X_1 = x_0 * V \\ X_2 = x_0 * \mu * M'_{12} * V \\ X_3 = 0. \end{cases} \tag{13}$$

Where $V = (I - \mu * M'_{11})^{-1} * P(q)$.

From the formula (13), we can conclude that the final feedback pages for a given query q belong to the set S_1 and S_2 only when $\nu \ll 1$ is true.

So, we can set $S = S_1 \cup S_2$.

4 Experimental

4.1 Experimental Setup

We construct experiment data set in order to verify the retrieval method of our approach described in Section 3.

The experiment data set is constructed by using the TREC WT10g test collection, which contains about 1.69 million Web pages (http://trec.nist.gov/),100 queries and the links among all pages. Stop words have been eliminated from all web pages in the collection based on the stop-word list and stemming has been performed using Porter Stemmer. [14]

4.2 Constructing the Test Data-Set

We Select all 100 queries $q_i, (i \in [1, 100])$ respectively. For each of the query q, we do the steps just as fellows:

(1) Computing the relevant value of the query q to all pages in WT10g; Selecting the $Top - N, (N = 500, 1000, 5000, 10^4, 1.5 * 10^4, 3 * 10^4)$ pages to construct

the data set $S_{1i}, (i = [1, 6])$ respectively. In order to illustrate the method, we assume the relevant value is TFIDF. For example,

$$S_{16} = \{t_i | tfidf(q, d_i) > 0 \wedge tfidf_1 \geq tfidf_2 \geq \cdots \geq tfidf_n > 0, i \in [1, 30000]\};$$

(2) Drawing up all links, which the source page is belonged to S_{1i}, to construct the link set $L_{1i}, (i = [1, 6])$.

$$L_{1m} = \{l_{ij} | \exists link(t_i \rightarrow t_j), t_i \in S_{1m}\};$$

(3) Constructing the data set $S_{2i}, (i = [1, 6])$ by collecting the target pages of links appeared in L_{1i}. If the page appeared in the S_{1i} and S_{2i} , it will be deleted from S_{2i}.

$$S_{2m} = \{t_j | link(t_i \rightarrow t_j) \in L_{1m} \wedge t_i \in S_{1m} \wedge t_j \notin S_{1m}\}.$$

(4) Combining S_{1i} and S_{2i}.

$$S_i = S_{1i} \cup S_{2i}, i \in [1, 6].$$

4.3 Irreducible and Aperiodic

The transition probability matrix has the property just as,

(1) Irreducible
According to the consist of the transfer matrix T, we can get $T = q \cup S_1 \cup S_2$. The links just as fellow are always exist.
- Link $q_i \rightarrow t_i, t_i \in S_1$;
- Link $t_i \rightarrow q_i, t_i \in S_1$;
- Link $t_i \rightarrow u_j, t_i \in S_1 \wedge u_j \in S_2$;
- Link $u_j \rightarrow q_i, u_j \in S_2$;

That means we can reach each other pages from one of the pages in set S. So, the matrix T is irreducible.

(2) Aperiodic
In the matrix T, all elements in the diagonal are positive except the $t_{00} = 0$. $T_{ii} > 0, (i \in (0, N])$ is always true. So, the matrix T is aperiodic.

So, the transition probability matrix T has a real, positive, and the biggest eigenvalue.

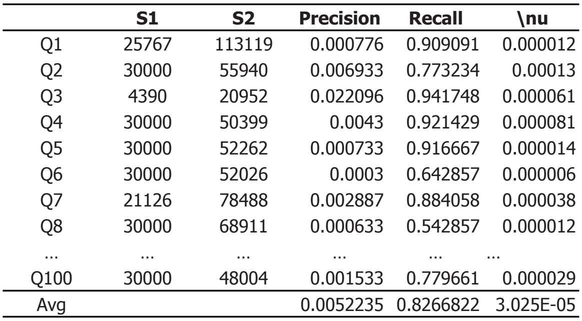
4.4 Experiment Results

In the follow description, we show the results when we deal with the top $3 * 10^4$ pages of `TFIDF`.

(1) ν
We use the formula (13) to compute the new pagerank only when $\nu \ll 1$ is true.

We define the ν as the sum of probability of page i when it is not belong to S_1, but it is indeed relevant to query q. The average ν is shown in Table 1.The result shows that $\nu = 3.025 * 10^{-5} \ll 1$ is true. So, we can compute the new pagerank by using formula (13).

Table 1. Result of ν for 30000 pages



	S1	S2	Precision	Recall	\nu
Q1	25767	113119	0.000776	0.909091	0.000012
Q2	30000	55940	0.006933	0.773234	0.00013
Q3	4390	20952	0.022096	0.941748	0.000061
Q4	30000	50399	0.0043	0.921429	0.000081
Q5	30000	52262	0.000733	0.916667	0.000014
Q6	30000	52026	0.0003	0.642857	0.000006
Q7	21126	78488	0.002887	0.884058	0.000038
Q8	30000	68911	0.000633	0.542857	0.000012
...	...	...	...	...	...
Q100	30000	48004	0.001533	0.779661	0.000029
Avg			0.0052235	0.8266822	3.025E-05

(2) Number(Relevance page)/ Number(All page)
We compute the new pagerank according to the formula (13) and sort them decreasingly. The ratio of number of relevance pages in top-N pages to the all the top-N pages can be computed.

The result is showed in Fig.3. We can find that the value of `TFIDF` is the biggest. The CLBCRA value increases with the increase of the number of pages in the data-set.

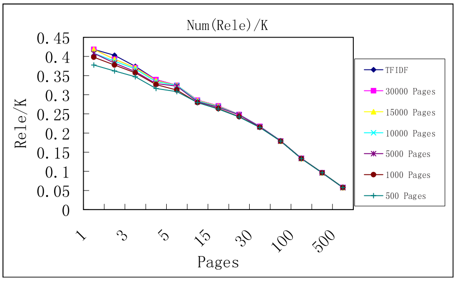


Fig. 3. Relevance pages / All pages

(3) Discussing the result
Considering the formula (13), we can find the solution of x_0,X_1,and X_2 are depended on V, where $V = (I - \mu * M'_{11})^{-1} * P(q)$.

If the $(\mu * M'_{11})$ is small, the V and the $P(q)$ will be very similar. As the M'_{11} is the truth, the μ decides the value of V. With the changing of μ, we can get different values of V.

In our experiment, because the μ is small, the final solution is similar to the $P(q)$, the `TFIDF` value.

5 Conclusion

This paper introduces two kinds of methods of information retrieval on the web, based on the hyper-link and based on the content. Both of them have shortages, such as the quantity of computation and the precision of retrieval, etc.

This paper draws a new framework by combining the `TFIDF` and `Pagerank` in order to support the precise results to users. We set up the computation model and get the final solution. We test the framework by using `TREC WT10g` test collection. The result shows that the precision of new method approaches the `TFIDF`'s. But the new framework has less quantity of computation than `TFIDF`.

However, in order to satisfy the users' actual information need, it is more important to find relevant web page from the enormous web space. Therefore, we plan to address the technique to provide users with personalized information.

Acknowledgements

This work was supported by project 06JK300 and 2005F08 of Shaanxi Province, P.R.China.

References

1. Raghavan, S., Garcia-Molina, H.: Complex queries over web repositories. In: VLDB 2003. Proceedings of 29th International Conference on Very Large Data Bases, pp. 33–44. Morgan Kaufmann, Berlin, Germany (2004)
2. Delort, J.-Y., Bouchon-Meunier, B., Rifqi, M.: Enhanced web document summarization using hyperlinks. In: HYPERTEXT 2003. Proceedings of the 14th ACM conference on Hypertext and hypermedia, pp. 208–215. ACM Press, New York (2003)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the 7th international conference on World Wide Web, pp. 107–117 (1998)
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
5. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: SODA '98. Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA. Society for Industrial and Applied Mathematics, pp. 668–677 (1998)
6. Steinberger, R., Pouliquen, B., Hagman, J.: Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 415–424. Springer, Heidelberg (2002)
7. Guo, G., Wang, H., Bell, D.A., Bi, Y., Greer, K.: An knn model-based approach and its application in text categorization. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 559–570. Springer, Heidelberg (2004)
8. Soucy, P., Mineau, G.W.: Beyond tfidf weighting for text categorization in the vector space model. In: IJCAI-05. Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, July 30-August 5, 2005, pp. 1130–1135 (2005)

9. Tal-Ezer, H.: Faults of pagerank / something is wrong with google mathematical model (2005)
10. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: VLDB 2004. Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3, pp. 576–587 (2004)
11. Podnar, I., Luu, T., Rajman, M., Klemm, F., Aberer, K.: A peer-to-peer architecture for information retrieval across digital library collections. In: Gonzalo, J., Thanos, C., Verdejo, M.F., Carrasco, R.C. (eds.) ECDL 2006. LNCS, vol. 4172, pp. 14–25. Springer, Heidelberg (2006)
12. Buntine, W.L., Aberer, K., Podnar, I., Rajman, M.: Opportunities from open source search. In: Skowron, A., Agrawal, R., Luck, M., Yamaguchi, T., Morizet-Mahoudeaux, P., Liu, J., Zhong, N. (eds.) Web Intelligence, pp. 2–8. IEEE Computer Society Press, Los Alamitos (2005)
13. Aberer, K., Klemm, F., Rajman, M., Wu, J.: An architecture for peer-to-peer information retrieval. In: Callan, J., Fuhr, N., Nejdl, W. (eds.) Workshop on Peer-to-Peer Information Retrieval (2004)
14. Sugiyama, K., Hatano, K., Yoshikawa, M., Uemura, S.: Improvement in tf-idf scheme for web peges based on the contents of their hyperlinked neighboring pages. Syst. Comput. Japan 36(14), 56–68 (2005)