NewPR-Combining TFIDF with Pagerank

Hao-ming Wang[1,2], Martin Rajman[2], Ye Guo[3], and Bo-qin Feng[1]

[1] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China
{wanghm, bqfeng}@mail.xjtu.edu.cn
[2] School of I & C, Swiss Federal Institute of Technology(EPFL), 1015 Lausanne, Switzerland
Martin.Rajman@epfl.ch
[3] School of Information, Xi'an University of Finance & Economics, Xi'an, Shaanxi 710061, P.R. China
guoyexinxi@126.com

Abstract. TFIDF was widely used in IR system based on the vector space model (VSM). Pagerank was used in systems based on hyper-link structure such as Google. It was necessary to develop a technique combining the advantages of two systems. In this paper, we drew up a framework by using the content of web pages and the out-link information synchronously. We set up a matrix M, which composed of out-link information and the relevant value of web pages with the given query. The relevant value was denoted by TFIDF. We got the NewPR (New Pagerank) by solving the equation with the coefficient M. Experimental results showed that more pages, which were more important both in content and hyper-link sides, were selected.

1 Introduction

With information proliferate 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, 2]

As a user, in order to find, collect and maintenance the information, which maybe useful for the specific aims, s/he has to pay more time, money and attention on the retrieval course.

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. For most of the users, they may not express their needs clearly with a few keywords. Users may be just interested in "most qualified" information or one peculiar part of returned information.

The search engines are based on one of the two methods, the content of the pages and the link structure.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 932–942, 2006.

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 are 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.

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

In this paper, we combine the relevance and the Pagerank of the web page in order to refine the retrieval results. We compute the `TFIDF` value firstly. And then, we compute the new Pagerank by the `TFIDF` and the out-link information of every page. The new Pagerank is called `NewPR`.

This paper is organized as follows: Section 2 introduces the concept of Pagerank and `TFIDF`. Section 3 describes the algorithm of `NewPR`. 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 Basic Concept

2.1 Pagerank

The Google search engine is based on the popular Pagerank algorithm first introduced by Brin and Page in Ref. [3].

Considering the pages and the links as a graph $G = P(Page, Link)$, we can describe the graph by using the adjacency matrix. The entries of the matrix, for example p_{ij}, can be defined as:

$$p_{ij} = \begin{cases} 1 & \exists\, Link(i \to j) \\ 0 & Otherwise. \end{cases}$$

Here $i, j \in (1, n)$ and n is the number of web pages. Because the total probability from one page to others can be considered 1, the rows, which correspond to pages with a non-zero number of out-links $deg(i) > 0$, can be made row-stochastic (row entries non-negative and sum to 1) by setting $p_{ij} = p_{ij}/deg(i)$. That means if the page u has m out-links, the probability of following each of out-links is $1/m$. We assume all the m out-links from page u have the similar probability.

For a real adjacency matrix P, in fact, there are many special pages without any out-link, which are called *dangling page*. Any other pages can reach the dangling page in $n(n \geq 1)$ steps, but it is impossible to get out. In the adjacency matrix, the row, corresponding to the dangling page is all zeros. Thus, the matrix P is not a row-stochastic. It should be deal with in order to meet the requirement of the row-stochastic.

One of the ways to overcome this difficulty is to change the transition matrix P slightly. We can replace the rows, all of the zeros, with $v = (1/n)e^T$, where

e^T is the row vector of all $1s$ and n is the number of pages of P contains. The P will be changed to $P' = P + d \cdot v^T$. Where

$$d = \begin{cases} 1 & if \quad deg(i) = 0 \\ 0 & Otherwise. \end{cases}$$

is the dangling page indictor [4]. If there were a page without any out-link from it, we could assume it can link to every other pages in P with the same probability. After that there is not row with all 0s in matrix P'.

P' is row-stochastic and it corresponds to the stochastic transition matrix over the graph G. Pagerank can be viewed as the stationary probability distribution over pages induced by a random walk on the web. It can be defined as a limiting solution of the iterative process.

Because of the existing of zero entries in the matrix P', it cannot guarantee the existence of the stationary vector. The problem comes from that the P' may be reducible. In order to solve the problem, P' can be modified by adding the connection between every pair of pages [4].

$$Q = P'' = cP' + (1-c)ev^T, \quad e = (1, 1, \cdots, 1)^T.$$

Where c is called dangling factor, and $c \in (0, 1)$. In most of the references, the c is set [0.85,1). [3]

After that, the Q is irreducible because all of the pages are connected (strong connection). For $Q_{ii}^{(k)} > 0, (i, k \in (1, n))$, the Q is aperiodic too. The Perron-Frobenius theorem guarantees the equation $x^{(k+1)} = Q^T x^{(k)}$ (for the eigensystem $Q^T x = x$) converges to the principal eigenvector with eigenvalue 1, and there is a real, positive, and the biggest eigenvector. [5,6]

2.2 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. [7]

Assuming vector $\tilde{d} = (d^{(1)}, d^{(2)}, ..., d^{|F|})$ represents the document d in a vector space. Each dimension of the vector space represents a word selected by the feature selection. The value of the vector element $d^{(i)} (i \in [1, |F|])$ is calculated as a combination of the statistics $TF(w, d)$ and $DF(w)$.

$TF(w, d)$ is the number of the word w occurred in document d. $DF(w)$ is the number of documents in which the word w occurred at least once time. The $IDF(w)$ can be calculated as

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

Where N_{all} is the total number of documents. The value $d^{(i)}$ of feature w_i for the document d is then calculated as $d^{(i)} = TF(w_i, d) \times IDF(w_i)$. Where $d^{(i)}$ is called the weight of word w_i in document d. [7]

The TFIDF algorithm learns a class model by combining document vectors into a prototype vector $\tilde{C}$ for every class $C \in$ Ç. Prototype vectors are generated by adding the document vectors of all documents in the class.

$$\tilde{C} = \sum_{d \in C} \tilde{d}.$$

This model can be used to classify a new document d'. Assuming vector $\tilde{d'}$ represents d', the cosine distance between $\tilde{d'}$ and $\tilde{C}$ is calculated. The d' is belonged to the class with which the cosine distance has the highest value.

3 Algorithm of the NewPR

3.1 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.1, we can get,

$$Precision = \frac{B}{Ret}; \quad Recall = \frac{B}{Ref}; \quad \gamma = \frac{Ref}{A+B+C+D} = \frac{Ref}{N}.$$

For a given retrieval system, the average value of *precision* and γ can be estimated. As the N is very large, γ is expected to be very small.

3.2 Page Link

We donate the query from the user with Q, all of the pages selected by retrieval system relevant to Q with $Y = \{y_i, i \in (1, n)\}$. The probability from Q to Y is $P = \{p_i, i \in (1, n)\}$, and from y_i returns to Q is $1 - \pi$. In our experiment, P is the TFIDF values of Q to Y.

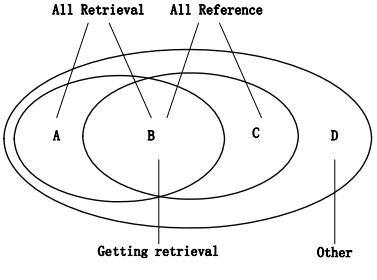


Fig.1 Concept of Information Retrieval

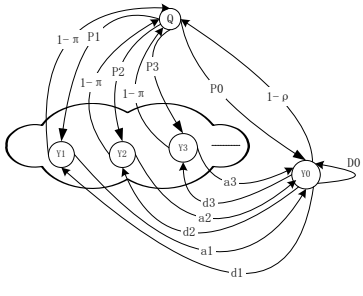
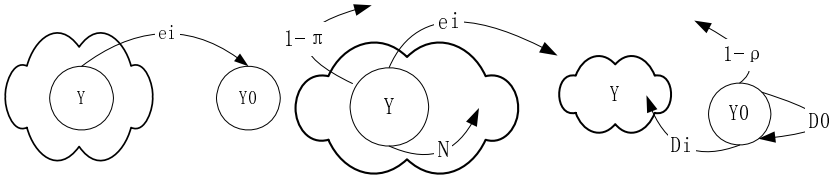


Fig.2 Information of Links



(A) From Y to Y_0 (B) From Y to others (C) From Y_0 to others

Fig. 3. Link Information of each page

We assume all the pages, which are not included in set Y, are included in a set Y_0. The probability of Y_0 transfers to itself is D_0, to Y is $D = \{d_i, i \in (1,n)\}$ and to Q is $1-\rho$. p_0 is probability from Q to Y_0. It is the sum of `TFIDF` values of Q to Y_0. The link information is showed in Fig.2.

Because the return link from Y to Q means the page belonged to part A in Fig.1, the probability $1-\pi = \dfrac{A}{Ret} = \dfrac{A+B-B}{Ret} = \dfrac{Ret-B}{Ret} = 1-Precision. \Rightarrow \pi = Precision.$

For the Q, assuming s_i is the `TFIDF` value, we get,

$$p_0 + \sum_{i=1}^{n} p_i = 1, \quad p_0 = \beta \sum_{i \notin (1,n)} s_i, \quad p_i = \beta s_i \Rightarrow \beta \sum_{i \notin (1,n)} s_i + \beta \sum_{i \in (1,n)} s_i = 1$$

$$\Rightarrow \quad \beta = \frac{1}{\sum\limits_{i \in ALL} s_i}, \quad p_0 = 1 - \frac{\sum\limits_{i \in (1,n)} s_i}{\sum\limits_{i \in ALL} s_i}, \quad p_i = \frac{s_i}{\sum\limits_{i \in ALL} s_i}. \quad (1)$$

In Fig.3(A), we assume the probability of page $y_i \in Y$ points to Y_0 is $e_i = \sum\limits_{j \notin Ret} n_{ij}$, where n_{ij} is the initial probability that page i points to page j.

In Fig.3(B), the page $y_i \in Y$ has three kinds of links: links to Q, links to Y, and links to Y_0. Thus, we get $(1-\pi) + e_i + \sum\limits_{j \in Ret} n_{ij} = 1$.

We define the link matrix $U = \{u_{ij} | i, j \in (1,n)\}$ as,

$$u_{ij} = \begin{cases} u_{ii} = 1 & \sum\limits_j u_{ij} = 0 \quad Dangling \quad page \\ 1 & \sum\limits_j u_{ij} > 0 \quad and \exists link\,(i \to j) \\ 0 & \sum\limits_j u_{ij} > 0 \quad and \nexists link\,(i \to j). \end{cases}$$

For the Y, we get

$$(1-\pi) + \sum_{j \in Ret} \beta u_{ij} + \sum_{j \notin Ret} \beta u_{ij} = (1-\pi) + \beta \sum_{j \in ALL} u_{ij} = 1$$

$$\Rightarrow \quad \begin{cases} \beta & = \dfrac{\pi}{\sum\limits_{j \in ALL} u_{ij}} = \dfrac{\pi}{OutlinkNum(i)}, \quad n_{ij} = \pi \dfrac{u_{ij}}{OutlinkNum(i)} \\ e_i & = \pi(1 - \dfrac{\sum\limits_{j \in Ret} u_{ij}}{OutlinkNum(i)}) = \pi(\dfrac{\sum\limits_{j \notin Ret} u_{ij}}{OutlinkNum(i)}). \end{cases} \tag{2}$$

For the Y_0, which is showed in Fig.3(C), we get

$$\rho = \frac{C}{C+D} = \frac{Ref - B}{N - Ret} = \frac{\gamma N - \pi Ret}{N - Ret} \approx \frac{\gamma N - \pi Ret}{N} \approx \gamma.$$

$$(1-\rho) + D_0 + \sum_{i \in Ret} d_i = (1-\rho) + \beta \sum_{j \notin Ret} \sum_{i \notin Ret} u_{ji} + \beta \sum_{j \notin Ret} \sum_{i \in Ret} u_{ji} = 1$$

$$\Rightarrow \quad D_0 = \rho \frac{\sum\limits_{j \notin Ret} \sum\limits_{i \notin Ret} u_{ji}}{\sum\limits_{j \notin Ret} OutlinkNum(j)} \quad , \quad d_i = \rho \frac{\sum\limits_{j \notin Ret, i \in Ret} u_{ji}}{\sum\limits_{j \notin Ret} OutlinkNum(j)}. \tag{3}$$

3.3 The Link Matrix

We assume the links among the pages in set Y composed the link matrix U.

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{pmatrix} \quad \widetilde{\mathbf{U}} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} & AA_1 \\ u_{21} & u_{22} & \dots & u_{2n} & AA_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} & AA_n \\ BB_1 & BB_2 & \dots & BB_n & BB_0 \end{pmatrix}$$

Adding the set Y_0 , U changes to $\widetilde{U}$. Where

$$AA_i = \sum_{j \notin Ret} u_{ij}, \quad BB_i = \sum_{j \notin Ret, i \in Ret} u_{ji}, \quad BB_0 = \sum_{i,j \notin Ret} u_{ij}.$$

We normalize the $\widetilde{U}$ by

$$\widetilde{m_{ij}} = \frac{\widetilde{u_{ij}}}{\sum\limits_j \widetilde{u_{ij}}} \quad i \in (1,n]\,; \quad \widetilde{a_i} = \frac{\widetilde{u_{ij}}}{\sum\limits_j \widetilde{u_{ij}}} \quad i \in (n, ALL)\,;$$

$$b_i = \frac{\sum\limits_{j \notin Ret, i \in Ret} u_{ji}}{\sum\limits_{j \notin Ret} OutlinkNum(j)}\,; \quad B_0 = \frac{\sum\limits_{j \notin Ret} \sum\limits_{i \notin Ret} u_{ji}}{\sum\limits_{j \notin Ret} OutlinkNum(j)}\,.$$

Adding the query , we get the transfer matrix T,

$$\mathbf{T} = \begin{pmatrix} 0 & p_1 & p_2 & \dots & p_n & p_0 \\ 1-\pi & n_{11} & n_{12} & \dots & n_{1n} & e_1 \\ 1-\pi & n_{21} & n_{22} & \dots & n_{2n} & e_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1-\pi & n_{n1} & n_{n2} & \dots & n_{nn} & e_n \\ 1-\rho & d_1 & d_2 & \dots & d_n & D_0 \end{pmatrix} = \begin{pmatrix} 0 & P & p_0 \\ 1-\pi & \pi M & \pi A \\ 1-\rho & \rho B & \rho B_0 \end{pmatrix}.$$

Where $A = (a_1, a_2, \cdots, a_n)'$,$B = (b_1, b_2, \cdots, b_n)$. P is the normalized value of TFIDF of Q to page $y_i (y_i \in Y)$. p_0 is the sum of normalized value of TFIDF of Q to pages in Y_0.

For a giving retrieval system, we could compute the B_0 and $B_i (i \in (1, n))$. We can get

$$\mathbf{T}' = \begin{pmatrix} 0 & 1-\pi & 1-\rho \\ P' & \pi M' & \rho B \\ p_0 & \pi A' & \rho B_0 \end{pmatrix}.$$

3.4 Computing Equation

From the equation $T'X = X$, we can get,

$$\begin{pmatrix} 0 & 1-\pi & 1-\rho \\ P' & \pi M' & \rho B \\ p_0 & \pi A' & \rho B_0 \end{pmatrix} \begin{pmatrix} x_0 \\ Y \\ y_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ Y \\ y_0 \end{pmatrix}$$

$$x_0 = (1-\pi)\|Y\|_1 + (1-\rho)y_0 \tag{4}$$
$$Y = x_0 P' + \pi M' Y + \rho y_0 B' \tag{5}$$
$$y_0 = x_0 p_0 + \pi A' Y + \rho B_0 y_0 \tag{6}$$
$$x_0 + \|Y\|_1 + y_0 = 1 \tag{7}$$

As the T is stochastic matrix, we get (7).

Changing (6), we get,

$$y_0 = \frac{x_0}{1-\rho B_0} p_0 + \frac{\pi}{1-\rho B_0} A' Y \,. \tag{8}$$

Changing (5), we get,

$$(I - \pi M' - \frac{\rho\pi}{1-\rho B_0} B' A') Y = x_0 (P' + \frac{\rho p_0}{1-\rho B_0} B') \,. \tag{9}$$

Assuming $C = \frac{\rho}{1-\rho B_0} B'$, we get,

$$y = x_0 [I - \pi(M' + CA')]^{-1} (P' + p_0 C) \,. \tag{10}$$

Assuming $V = [I - \pi(M' + CA')]^{-1}(P' + p_0 C)$, we get,

$$Y = x_0 V \Rightarrow \|Y\|_1 = x_0 \|V\|_1 \,. \tag{11}$$

Changing (4), we get,

$$[1 - (1-\pi)\|V\|_1] x_0 = (1-\rho) y_0$$
$$y_0 = \frac{1 - (1-\pi)\|V\|_1}{1-\rho} x_0 \,. \tag{12}$$

Combining the formula (7)(11)(12), we get

$$x_0 = \frac{1}{1 + \dfrac{1 + (\pi - \rho)\|V\|_1}{1-\rho}} \quad ; \quad y_0 = \frac{1 - (1-\pi)\|V\|_1}{1-\rho} x_0 \quad ; \quad Y = x_0 X \,. \tag{13}$$

4 Experimental

4.1 Experimental Setup

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

The experiment is constructed by using the `TREC WT10g` test collection, which contains about 1.69 million Web 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. [7]

(1) Selecting test pages. We construct the set R with all pages which are relevant to the query $q_i, i \in (1, 100)$. The data-set D can be set up just as

$$d_i = \begin{cases} d_i, & (d_i \in R); \\ d_j, & \exists\,(link(i \to j) \wedge link(j \to k)),\ (j \notin R; \quad i,k \in R) \\ d_j, d_l & \exists\,(link(i \to j) \wedge link(j \to l) \wedge link(l \to k)),\ (j,l \notin R; \quad i,k \in R)\,. \end{cases}$$

We name all pages in D from 1 to 12486 and pick up all out-links from those pages.

(2) Computing the old Pagerank. In order to compare the result of new method with the traditional one, we compute the pagerank of the every page in traditional way firstly. In this method, we ignore the last column of link matrix P , and it guarantee the link matrix is square one.

It must be noticed that the pagerank value of pages in our experiment are not very precise. The reason is that we consider the link information of pages belonged to the data set D only. There may be many important links out of the D have not be considered. Table. 1 shows the top 10 results of pagerank according to the traditional method.

(3) Computing the NewPR. We compute the NewPR by using *Matlab* with the parameter of link matrix P. The formula (1)(2)(3)(13)have been mentioned above. In the program, we assume the two parameters $\pi = 0.6$ and $\rho = 0.1$. Table. 3 shows the NewPR of the query 511. The detail of this query can be checked in `WT10g`. Due to the capability of the computer, we compute the first 5000 pages.

4.2 Experiment Results

In order to compare the two methods, the OldPR and the NewPR, we need to consider two questions, (1) Are the NewPR and the OldPR similar? (2) Is the NewPR better than OldPR?

We can compute the Spearman Rank Correlation Coefficient in order to determine the difference between the OldPR and the NewPR. The Spearman Rank Correlation Coefficient is defined by

$$r' = 1 - 6 \sum \frac{d^2}{N(N^2 - 1)}.$$

Table 1. OldPagerank

Rank	Page_No	Old_Value
1	3971	10.00000
2	3973	10.00000
3	3976	10.00000
4	3682	5.91328
5	3897	5.47881
6	3898	5.47881
7	3901	5.47881
8	1104	5.28466
9	1664	4.89549
10	1396	4.81016
...	...	...

Table 2. TFIDF

Rank	Page_No	TFIDF-Q511
1	1798	16.52122
2	1805	16.40058
3	7280	16.28200
4	1609	16.20084
5	1787	16.09591
6	11459	15.90077
7	1780	15.85325
8	1851	15.83046
9	1745	15.80058
10	11443	15.76907
...	...	...

Table 3. NewPR

Rank	Page_No	NewPR
1	3560	0.64832
2	2597	0.47899
3	3553	0.46598
4	3558	0.41606
5	3559	0.40520
6	1673	0.39539
7	2857	0.35789
8	4848	0.31863
9	1776	0.30918
10	1790	0.29264
...	...	...

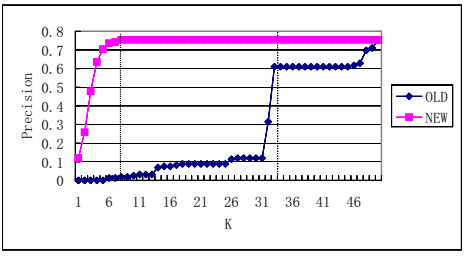
Table 4. Rank

Page_No	Old_Rank	New_Rank	d^2
1	1972	1319	426409
2	949	2283	1779556
3	1973	2304	109561
4	1974	2150	30976
5	4625	2203	5866084
6	1975	2216	58081
7	1976	2243	71289
8	4798	2255	6466849
...	...	...	...

Table 5. Precision

K	Old_Num of Rele	New_Num of Rele	Old_Prec	New_Prec
100	0	19	0.00000	0.11950
200	0	41	0.00000	0.25786
300	0	76	0.00000	0.47799
400	0	101	0.00000	0.63522
500	0	112	0.00000	0.70440
600	2	117	0.01258	0.73585
700	2	118	0.01258	0.74214
800	3	120	0.01887	0.75472
...	...	...	...	...

Table 6. Feedback Rele/AllRele



Where N is the number of total pages, and d is the difference in statistical rank of corresponding variables, and $r' \in [-1, +1]$. $r' = 0$ means that there is no correlation between the two quantities. They are completely independent of one another. Table.4 shows the Old_Rank, New_Rank and the d^2. We can compute $r' = 0.0046$ of all 5000 pages. That means the two algorithms, OldPR and NewPR are almost independent. This result answers the first question.

For the second question, we check the first top 100, 200, $\cdots$, 5000 pages of two methods, calculate the number of pages related to the query 511. In order to

compare the speeds of two methods' of reaching the maximal number of relevance pages, we compute the *precision*, ratio of relevance pages in the feedback pages list over all relevance pages. The result is showed in Table.5. From Table.6, we can find that the speed of new method is faster than that of old one. In the new method, it reach the top value in about 800 pages, meanwhile it needs almost all 5000 pages in the old method.

5 Conclusion

This paper introduces the methods of information retrieval on the web, and the concept of `TFIDF` and `Pagerank`. Due to the different methods of these two kinds of technologies use, the `TFIDF` cannot reflect the link information among pages. Meanwhile the `Pagerank` does not consider the content of pages.

We draw up a new framework by combining the `TFIDF` and `Pagerank` in order to support the precise results to users. We test the framework by using `TREC WT10g` test collection. The experimental result shows that the new method gives a better effect. But we find that the effect is not so distinct, we want to consider the in-link of every page in the future. In other side, we should change the value of α, which affects the final result of page order.

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 *2004F06* and *2005F08*, Research of Nature Science of *Shaanxi Province, P.R.China*.

References

1. Raghavan, S., Garcia-Molina, H.: Complex queries over web repositories. In: Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003), September 9-12, Berlin, Germany, Morgan Kaufmann (2003) 33–44
2. Delort, J.Y., Bouchon-Meunier, B., Rifqi, M.: Enhanced web document summarization using hyperlinks. In: Proceedings of the 14th ACM conference on Hypertext and hypermedia(HYPERTEXT 2003), ACM Press (2003) 208–215
3. 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)
4. Bianchini, M., Gori, M., Scarselli, F.: Inside pagerank. ACM Transactions on Internet Technology **5**(1) (2005) 92–128
5. Eiron, N., McCurley, K.S., Tomlin, J.A.: Ranking the web frontier. In: Proceedings of the 13th international conference on WWW2004, ACM Press (2004) 309–318

6. Boldi, P., Santini, M., Vigna, S.: Pagerank as a function of the damping factor. In: Proceedings of the 14th international conference on World Wide Web(WWW 2005), ACM Press (2005) 557–566
7. 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) (2005) 56–68