EFFECTIVELY FINDING THE RELEVANT WEB PAGES FROM THE WORLD WEB WIDE

Dr.N.P.Gopalan[1], Prof. J. Akilandeswari[2], P. Gabriel Sagaya Selvam[3]
HOD I/C, Asst.Prof/Dept of CSE[1], Asst. Prof/Dept of CSE[2], II yr M.E (CSE)[3]
National Institute of Technology[1], Sona College of Technology[2], Sona College of Technology[3]
National Institute of Technology[1], Anna University[2], Anna Univerisity[3]
Trichy – 620015[1],Salem – 636005[2],Salem – 636005[3]
India[1], India[2], India[3]

Abstract:

Effective & efficient retrieval of the required quality web pages on the web is becoming a greater challenge. Early work on search engines concentrated on the textual content of web pages to find relevant pages, but in recent years, the analysis of information encoded in hyperlinks has been used to improve search engine performance.

For these reasons, this paper presents three hyperlink analysis-based algorithms to find relevant pages for a given web page (URL). The first algorithm comes from the extended co-citation analysis of the web pages. The second one takes advantage of linear algebra theories to reveal deeper relationships among the web pages and to identify relevant pages more precisely and effectively. The third one presents a variation on the use of linkage analysis for automatically categorizing web pages, by defining a similarity measure. This measure is used to categorize hyperlinks themselves, rather than web pages. Also this paper presents a moved Page Algorithm to detect and eliminate the dead pages.

Key-words: Hyperlink analysis, Singular Valued Decomposition, Page Source, Relevant, Moved Pages, Similar search

1. Introduction

The World Wide Web is a rich source of information and continues to expand in size and complexity. How to efficiently and effectively retrieve required Web pages on the Web is becoming a challenge. A relevant Web page is the one that addresses the same topic as the original page, but is not necessarily semantically identical. Providing relevant pages for a searched Web page would prevent users from formulating new queries for which the search engine may return many undesired pages. By definition, Relevancy is the one that addresses same topic as original page, but is not necessarily semantically identical one.

Hyperlink has its own advantages where it encodes a considerable amount of latent human judgment in most cases. The creators of web pages create links to other pages, usually with an idea in mind that the linked pages are relevant to the linking pages. Therefore, a hyperlink, if it is reasonable, reflects the human semantic judgment and judgment is objective and independent of synonymy, polysemy of the words in the pages. When hyperlink analysis is applied to the relevant page finding, its success depends on how to solve the following two problems:

1) How to construct a page source that is related to the given page.

2) How to establish effective algorithms to find relevant pages from page source.

2. Problem Formulation

The Co-citation algorithm, Companion Algorithm, Page Rank Algorithm and Kleinberg's HITS Algorithm finds relevant web pages from the Hyperlink structure. They are all relevant to the given URL in a broad sense, rather than in a semantic sense. The hyperlink, because it usually conveys semantics between the pages, has attracted much research interest. In Kleinberg's work, which applies the HITS algorithm to find relevant pages, the page source is derived from the parents of the given page, i.e., the page source consists of parent pages and those pages that point to, or are pointed to by, the parent pages. However, since the pages pointing to the parents connect to the given page via two-level hyperlinks and a Web page usually refers to multiple topics, they might have weak semantic relationships with the given page and, in turn, the page source might not be rich in related pages. Dean and Henzinger construct page source in a different way for their relevant page finding algorithm Companion. Their pages source consists of parent and child pages of the given page u, as well as those pages that are pointed to by the parent pages of u and those pages that point to the child pages of u. This page source construction is more reasonable as all the pages in the page source are at the same link level with the given page u and have close relationships with u.

The hyperlinks between the pages on the same host are omitted in this page source construction, which might filter some semantically relevant pages on the same host about certain topics. This page source construction does not consider intrinsic page treatment in the parent and child page sets of u, which might result in the algorithm being easily affected by malicious hyperlinks. Kleinberg applies his HITS algorithm and Dean and Henzinger apply their improved HITS algorithm to their own page source. Instead of finding relevant pages from page similarities, they find authority pages as relevant ones from mutual page relationships that are conveyed by hyperlinks.

3 Problem Solutions

Ideally, the page source, a page set from which the relevant pages are selected, should have the following properties:

1) Size of page source is relatively small
2) Page source is rich in relevant pages.

The page source for relevant page finding is derived directly from a set of parent pages of the given page. Whenever applying effective algorithm to page source, the top 10 authority pages with the highest authority weights are considered to be the relevant pages of the given page. If the page source is not constructed properly, i.e., there are many topic unrelated pages in the page source. Then the whole process of finding the relevant pages goes in vain.

The new page source, based on which the algorithms are established, is constructed with required properties. The page similarity analysis and definition are based on hyperlink information among the web pages. The first algorithm, Extended Co-citation algorithm **[1]**, is a co-citation algorithm that extends the traditional co-citation concepts. It is intuitive and concise. The second one, named Latent Linkage Information (LLI) algorithm **[1]**, finds relevant pages more effectively and precisely by using linear algebra theories, especially the singular value decomposition (SVD) **[3]** of matrix, to reveal deeper relationships among the pages. The third one, named Equivalent Hyperlink algorithm **[2]** finds relevant pages more effectively and precisely by finding correlation between the two hyperlinks in terms of the anchor text and the content of the web page referred to for each hyperlink.

In our proposed model, each of these hyperlinks is an instance mapping of the same hyperlink function, a function that maps from anchor text references to web pages. The benefit of this approach is that given a particular hyperlink, we can consider all the other hyperlinks representing mappings of the same function as leading to pages that are potentially similar to the one the current hyperlink maps to. The fourth one, named Finding Moved Pages algorithm **[4]**, which detects and eliminates the dead pages that is no longer accessible in the web.

3.1 Structure of this paper

The paper is organized as per procedure: First, the Extended Co-citation algorithm is discussed. Second, Latent Linkage Information algorithm (LLI) is discussed. Third, Equivalent Hyperlink algorithm is discussed and finally Finding Moved Pages algorithm is discussed. A study of comparison is made over the algorithms for an arbitrary page source constructed over the given URL ‘u’.

4. Algorithms Description

4.1 Extended Co citation Concept

For a pair of documents p and q, if they are both cited by a common document, documents p and q is said to be co cited. The number of documents that cite both p and q is referred to as the co citation degree of documents p and q. The similarity between two documents is measured by their **co citation degree**. Since there exists no pre known page source for given page and co citation analysis, the success of co citation analysis mainly depends on how to effectively construct a page source with respect to the given page. Meanwhile, constructed page source should be rich in related pages with a reasonable size as shown in **Fig 1** such that page u contains parent **(BS)** and child pages **(FS)**.

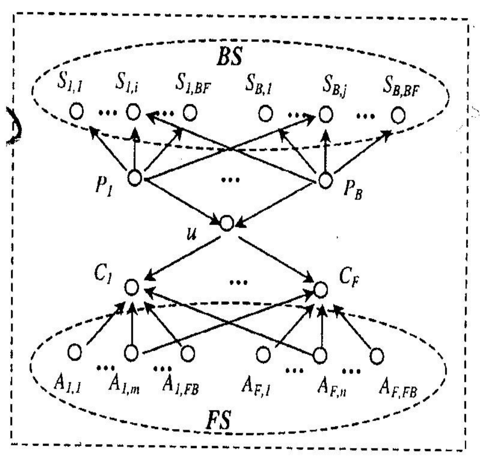

Fig 1. Page source structure

Extended Co citation algorithm overcomes defects of traditional co citation algorithm where child pages are not taken into account in page source construction. The page source is derived from parent and child pages of the given page. The in-view is a set of parent pages of U and out-view is set of child pages of u. Page source is constructed with the following.

1) Page u
2) Up to B parent pages of u and up to BF child pages of each parent page that is different from u.
3) Up to F child pages of u and up to FB parent pages of each child page that is different from u.

4.1.1 Extended Co citation Algorithm

1) Choose up to B arbitrary parents of u.
2) For each of these parents p, choose up to BF children of p that surround the link from p to u. Merge the intrinsic or near duplicate parent pages if they exist. Let P_U is set of parent pages of u.
Pu = {p_i / p_i is a parent page of U without intrinsic and near duplicate pages, i $\in$ [1,B]}, Let S_i = {$s_{i,k}$ / $s_{i,k}$ is child page of Pi, $s_{i,k}$=u, $P_i \in P_u$, k$\in$ [1,BF]}, i$\in$ [1, B].

Then steps 1, 2 produces following set:

$$\text{Thus BS} = \bigcup_{i=1}^{B} S_i$$

3) Choose first F children of U.
4) For each of these children c, choose up to FB parents of c with highest in-degree. Merge the intrinsic or near duplicate child pages if they exist. Let C_u be a set of child pages of u. C_u = {C_i / C_i is a child pages of u without intrinsic and near duplicate pages, i$\in$ [1,F]}. Let A_i = {$a_{i,k}$/$a_{i,k}$ is a parent page of page C_i, $a_{i,k}$ and u are neither intrinsic nor near duplicate page, Ci $\in$ Cu, K $\in$ [1,FB]};i$\in$[1,F]
Then steps 3, 4 produces following set:

$$\text{Thus FS} = \bigcup_{i=1}^{F} A_i$$

5) For a given selection threshold δ, select pages from BS and FS such that their back co citation degrees or forward co citation degree with u are greater than or equal to δ. These selected pages are **relevant pages** of U.
RP = {Pi / $P_i \in$ BS with b (p_i, u) ≥ δ or $p_i \in$ FS with and f (p_i, u) ≥ δ}

4.2 Latent Linkage Information (LLI)

Although Extended Co citation algorithm is simple and easy to implement, it is unable to reveal the deeper relationships among the pages. For example, if two pages have the same back (or forward) co citation degree with the given page u, the algorithm cannot tell which page is more relevant to u. For this, the **singular value**

decomposition (SVD) of a matrix in linear algebra has such properties that reveal the internal relationship among matrix elements **[5]**

4.2.1. SVD Background

The SVD of a matrix is defined as follow. Let $A = [aij]_{m*n}$ be a real m*n matrix. Without loss of generality, we suppose m>=n and the rank of A is rank (A)= r. Then, There exist orthogonal matrices U_{m*m} and V_{n*n} such that

$$A = U\begin{pmatrix} \sum \\ 0 \end{pmatrix}V^T = U\sum V^T \quad \text{------------------------} \quad (1)$$

Where $U^TU=Im; V^TV=In; \sum_1 =$ diag(σ1.σn) σiσi+1 > 0, for 1<=I<=r-1; σj=0 for j>= r+1, ∑ is an m*n matrix, U^T and V^T are the transpositions of matrices U and V respectively, Im and In represent m*m and n*n identity matrices, separately. The rank of A indicates the maximal number of independent rows or columns of A. **Equation (1)** is called the singular value decomposition of matrix A. The singular values of A are diagonal elements of ∑. The columns of U are called left singular vectors and those of V are called right singular vectors. Since the singular values of A are in non-increasing order, it is possible to choose a proper parameter k such that the last r-k singular values are much smaller than the first k singular values and these k singular values dominate the decomposition.

4.2.1 LLI Algorithm

If the size of BS is m and size of Pu is n, sizes of FS and Cu are p and q respectively. Without loss of generality, we also suppose m > n and p > q. The topological relationships between the pages in BS and Pu are expressed in a **linkage matrix** A, and the topological relationships between the pages in FS and Cu are expressed in another linkage matrix B. The linkage matrices A and B are concretely constructed as follows:

A =(aij) m*n;

$$Aij \begin{cases} = 1 \text{ when page i is child of page j;} \\ \quad \text{page i } \varepsilon \text{ BS; Page j } \varepsilon \text{ Pu;} \\ = 0 \text{ otherwise} \end{cases}$$

B = (bij) p*q;

$$Bij = \begin{cases} = 1 \text{ when page i is parent of page j;} \\ \quad \text{page i } \varepsilon \text{ FS; page j } \varepsilon\text{Cu;} \\ = 0 \text{ otherwise} \end{cases}$$

Since A and B are real matrices, there exist SVDs of A and B:

$$A = Um*m\sum m*nV^Tn*n,$$

$$B = Wp*p\ \Omega p*qX^Tq*q.$$

For the SVD of matrix A, matrices U and V can be denoted as $U_{m*m}=[u1, u2, \ldots um]_{m*m}$ and $Vn*n=[v1,v2\ldots vn]_{n*n}$; Where ui[i=1…m] is m-dimensional vector and $u_i = (u1i,u2i\ldots umi)^T$ and v_i [i=v1…n] is an n-dimensional vector vi $=(v1i,v2i\ldots.vni)^T$ Suppose rank(A)= r and singular values of matrix A are

σ1 >=σ2>=… >=σr>=σr+1…. = σn = 0:

For a given threshold ε (0 < ε< 1), we choose a parameter k such that $(\sigma_k - \sigma_{k+1})/\sigma_k >= \varepsilon$". Then, we denote $U_k = \{u1; u2;..;uk\}_{m*k}$, $V_k=(v1;v2;\ldots;vk)_{n*k}$, $\sum k=diag(\sigma1;\sigma2;..;\sigma_k)$ and $A_k=U_k\sum_k V_k^T$

The best approximation matrix Ak contains main linkage information among the pages and makes it possible to filter those irrelevant pages, which usually have fewer links to the parents of given u, and effectively finds relevant pages. In this algorithm, the relevance of a page to the given page u is measured by the similarity between them. For measuring the page similarity based on Ak, We choose the i^{th} row R_i of the matrix $U_k\sum_k$ as the coordinate vector of page i (page i ε BS) in a k-dimensional subspace S:

$Ri^1 = (wi_1w1; wi2w2;..;wilwl)$; i=1,2…p

The projection of coordinate vector u in the l-dimensional subspace L is represented as

$U^{11=}$ uXlΩ1=(g111; g_2^{11}; . . . ; $g_l^{11)}$ Where $g_l^{11} = \sum g_j X_{ji} w_i \quad {}_{i=1,2\ldots l}$

Therefore, the similarity between a page i in FS and the given page u is

$$FSSi = sim(Ri^1; u^{11}) = \frac{|Ri^{1*}u^{11}|}{\|Ri^1\|_2\|u^{11}\|_2}$$

For the given selection threshold δ, the relevant pages in FS with respect to given page u is set FSR = {pi | FSS_i>=δ; pi ε FS; i=1,2…p} Finally, **Relevant pages** of given page (URL) u is page set **RP**=BSR ∪ FSR.

4.3 Equivalent Hyperlink Concept

The web is considered to be a graph data structure in which pages form nodes and hyperlinks form edges between the nodes of the graph. An alternative view of hyperlinks is that of a function. Each hyperlink represents a mapping H from web pages and their anchors to web pages [2]

H :: (URL, Anchor) => URL

Given the URL of a web page and the identity of one of the anchors in the page, the function returns the URL of another web page.

4.3.1 Equivalent Hyperlink Algorithm

The algorithm works over hyperlinks such that given a hyperlink, in terms of its anchor text and target page, it produces a list of URLs representing the target pages of hyperlinks similar to input hyperlink **[2].**

Procedure EH ((S, a), T) {
Similar_to = FindSimilar(T)
Link_to={b/Vt similar_to Vb ε FindBackLinks(t)}
SimilarAnchor_to=FindSimilarAnchor(a)
SimilarLinks={t/∨t ε similar_to; ∨u ε link_to ∧ Similar_to; t ε Links (page (u)}
Return SimilarLinks
}

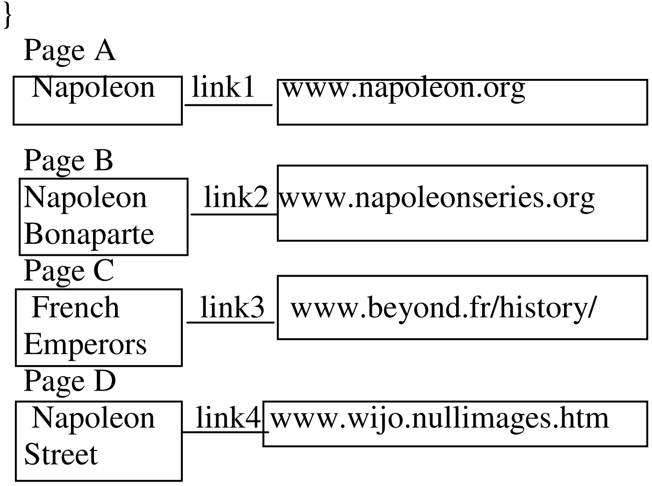

Fig2 Webpages, anchor & links for "Napoleon"

The input to the EH algorithm is hyperlink {(s,a);T} represented by the web pages and the anchor text 'a' in S and the target web page T.

The input is ((A, "Napoleon"), www.napoleon.org). The Steps taken by algorithm are as follows for **fig 2**.

1) **Calculate the set of the web pages similar to www.napoleon.org.** The function find similar is called on the web page www.napoleon.org to find web pages with similar content this set of web pages is denoted by Similar to www.napoleon.org,www.napoleonserios.org,www.heyond.fr/history/kings.html

2) **Find web pages linking to the set of similar web pages**. For each page in the set similar to find the web pages that link to it. Link_to denotes the set of all web pages. Result is {A, B, C}

3) **Find all pages with similar anchor text to "Napoleon"**. Function find similarAnchor is called to find all web pages containing an anchor with text similar to "Napoleon". Result is {A.B.D}.

4) **Find the Class reference URLs**. The URLs of pages that link to pages similar to www.napeoleon.org.

5) **Find Cross reference URL to** the link_to set of {A, B, C} and Similar Anchorset of {A, B, D} we obtain similar links set {A, B}.

4.4 Finding Moved Pages Algorithm

In 1995, it was found that 14.9% of the URLs returned by popular search engines no longer point to accessible pages. With the growth and aging of the web since their measurements, the percent of obsolete URLs returned may now be even higher. Currently, there are no utilities that try to track down moved pages. At the time the search was performed, U_{old} returned by search engine is no longer existed. Here we describe two approaches, using link-based heuristics **[4].** One approach termed as Exploiting Hyper link Structure is to automatically finding new URL U_{new} is based on the observation that the people most likely to know the new location of the page are those who cared enough about the material to point to the old location. The pages R that referenced U_{old} at the time they were last crawled could then check whether each page R pointed to the new location U_{new} of the moved page, either directly or recursively, preferentially expanding links whose anchor text included blurb terms. It would return a page to the user if the page matched the criteria of the original search and contained the information appearing in the original blurb.

5. Comparative Study

In our experiment, we selected an arbitrary Web page u="http://www.java.com" which is the home page of Java platform as the given page (URL). The AltaVista Web search engine **[6]** obtains the page source for this given page. For comparison, the Extended Co-citation algorithm, LLI algorithm, DH Algorithm, and Companion algorithm were applied to this page source. Based on the experimental results, algorithm comparison was conducted. A numerical experiment was also conducted on Extended Co-citation algorithm and the LLI algorithm to show that the LLI algorithm is able to reveal deeper relationships among the pages. First, we compare the LLI Algorithm and the Extended Co-citation algorithm based on their experiment results. We chose the top 10 returned relevant pages of each algorithm for comparison.

The results returned by the Extended Co-citation algorithm have more semantically relevant pages as 40 % of more relevant pages & the results returned by LLI algorithm have more semantically relevant pages as 60 % of more relevant pages. Since LLI reveals more deeper relationship among web pages. For Equivalence Hyperlink algorithm, if the anchor text provided for hyperlink is richer, the output is also more relevant web pages. Hence Equivalent Hyperlink algorithm heavily depends on rich anchor set provided. Whenever Moved pages algorithm is included, the output of relevancy is highly increased from the previous result set.

6. Conclusion

In this paper, we have proposed three algorithms to find relevant pages of a given page: the Extended Co-citation algorithm and Latent Linkage Information algorithm and Equivalent Hyperlink Algorithm **[1], [2].** Also we have proposed finding moved pages algorithm to detect and eliminate the dead pages found in the web. These three algorithms are based on hyperlink analysis among the pages and take a new approach to construct the page source. The new page source reduces the influence of the pages in the same website to a reasonable level in the page similarity measurement, avoids some useful information being omitted, and prevents the results from being distorted by malicious hyperlinks. These three algorithms could identify the pages that are relevant to the given page in a broad sense, as well as those pages that are semantically relevant to the given page. Furthermore, the LLI algorithm reveals deeper relationships among the pages and finds out relevant pages more precisely and effectively. The Equivalent hyperlink algorithm **[2]** is used to identify similar hyperlinks that are semantically similar. However it depends on quality of anchor text as well as search engine techniques to provide good lists. Also this paper presented a moved Page Algorithm to detect and eliminate the dead pages.

Our future works is investigating through XML documents and SSL web documents for further effectiveness in finding relevant pages. The ideas and techniques in this work would be helpful to other web related researches.

References:

[1] Jingyu Hou and Yanchun Zhang, "Effectively Finding Relevant Web Pages from Linkage Information, August 2003

[2] Simon Courtenage and Steven Williams, "Finding Relevant Web Pages Through Equivalent Hyperlinks", University of Westminster, Apr.2004.

[3] B.N. Datta, "Numerical Linear Algebra and Application". Brooks / Cole Publishing, 1995.

[4] Ellen Spertus," ParaSite: Mining Structural Information on the Web", University of Washington, August 2003.

[5] Dunham, Margaret.H, "Data Mining", Pearson Education Asia; 2003

[6] http://www.altavista.com/, AltaVista Search engine, 2003