From Content-based to Connectivity-based Search on the World Wide Web —
A Journey Trail

M M Sufyan Beg

Department of Electrical Engineering, Indian Institute of Technology, New Delhi, 110 016. mmsbeg@ee.iitd.ac.in

Searching the World Wide Web is an activity whose importance cannot be emphasized any further. While searching the web, a ranked list of all-and-only relevant documents is desired, in response to a query. The web search efforts started with the application of classical content-based information retrieval techniques to the web. However, owing to commercial pressures, spamming became a major threat to the success of content-based techniques. The work then diverted to exploit the hypertextual structure of the web. The number as well as quality of hyperlinks to a page is taken to be a measure of authority of that page. A page is more authoritative if many pages point to it, or alternatively, if some high-authority pages point to it. Such an authoritative value may directly be taken as a basis for ranking of several given pages. The approach helps in fighting the spamming problem as the importance of a given page is now gauged not by the contents of that very page alone, but rather by the (hypertextual) contents of other pages on the web. It gives an unbiased factor to the ranking procedure. Hypertextual structure has proved useful in finding the related pages. The quest for related pages is nothing but search-by-example. Instead of issuing a query, we come up with a sample page and search for pages similar to it. It helps in situations where formulation of a suitable query is difficult, particularly when the information space is unfamiliar to the user. If one models the web as a directed graph, with the web pages being the nodes and the hyperlinks between them being the edges of the model graph, then starting from a sample page, the breadth-first search to a few levels would give a set of pages that are related to the sample page.

1 Introduction

With the revolution that the emergence of Internet has brought to the domain of Information Technology, the web search has become an inevitable operation. Just as the World Wide Web has emerged as the global network, searching this World Wide Web has also become a global necessity. This web search activity may range from locating a focused research paper pertaining to an area of specialization, down to some reasonable material that could help in making an essay for a school boy. E-commerce applications, where discovering right resources means a boost in the income, make the web search that much more important. On the whole, it could easily be claimed that every single net-user needs to do web searching some times or the other. While searching the web, all of us wish to have all-and-only documents relevant to the query posed. It is also preferable to have this set of relevant documents listed in a decreasing order of relevance. Thus, a better web search system is the one, which, in response to a query, returns a ranked list of relevant documents.

The recent past has seen a shift in thrust from the conventional content based searching\(^1\)\(^2\) to the more crisp connectivity based searching\(^3\). A few years ago, the query term frequency was the single main heuristic in ranking the web pages. But the emergence of novel search engines like Google\(^4\) has marked the beginning of the era of connectivity based or citation based, or more commonly known as hyperlink based (or simply link based) web searching. It all started with the revolutionary work done by Kleinberg\(^5\), to be consolidated later by Brin and Page\(^6\), culminating in what is now famous as the Google search engine for the web. These works have primarily been concentrating on harnessing the additional information hidden in the hypertext structure of the web pages. Barring some occasional exceptions, a hyperlink on a given page \(p_i\) points to a page \(p_j\), which happens to lie in the same context as the page \(p_i\). The breadth-first search to a few levels would, therefore, give a set of pages that are related to the page \(p_i\). Moreover, the very fact that a page \(p_j\) is pointed to by many other pages, depicts the importance the page \(p_j\) holds in the views of the web.
community. Hence the typical web-authoring tendency to provide hyperlinks to the related pages gives rise to a new paradigm in web search activity. We shall highlight this shift of thrust in this paper.

We begin by providing in Section 2, a brief overview of the classical information retrieval (IR) techniques as applied to the web. In Section 3, we shall take up the techniques used in harnessing the hyperlink structure of the web for propelling the ranking algorithms. Section 4 would be exclusively devoted to the techniques for finding the related pages on the web. We shall conclude in Section 5.

2 Content-Based Web Search

Going by the instinct, the contents of a page are the best descriptors of that page. But to get the index terms of a given document is a non-too-trivial task. It is a well-known fact that for a good amount of time, Yahoo! relied on human intelligence to get this indexing done. Equivalently, we may have to employ the complex Natural Language Processing techniques to automate the job of indexing. However, some easier techniques have been researched upon by the IR community for this task of keyword selection. We review a few of them here. But before we proceed, we must have a look at the text pre-processing operations, which are a pre-requisite for the application of any of the content-based techniques.

2.1 Text Pre-Processing

First of all, we need to remove the stop-words, the words that have very little semantic content. They frequent homogeneously across the whole collection of documents. They are generally the prepositions or articles, like the, an, for, etc. Over a period of time, people have come up with a list of stop-words pertaining to a general domain. However, it may be argued that a stop-word is very much context dependant. A word like web may be treated as a stop word in a collection of web-related articles, but not so in a set of literary documents.

The text pre-processing also includes an important step of word stemming, wherein all the words with the same root are reduced to a single form. This is achieved by stripping each word of its suffix, prefix or infix, if any. That is to say that all the words are reduced to their canonical form. For instance, the words like borrow, borrower, borrowed and borrowing, all would be reduced to the stem word borrow. This way the burden of being very specific while forming the query, is taken off from the shoulders of the user. A well-known algorithm for carrying out word stemming is the Porter Stemmer algorithm.

It may be noted that the text pre-processing techniques are very much dependant on the language of the document. For instance, just the removal of suffixes may usually suffice as the stemming technique in the case of English language, but not necessarily so with other languages.

2.2 Term Frequency (TF)

The frequency of a word in a given document is taken as a good measure of the importance of that word in the given document. This, of course, holds true only after the text pre-processing has been carried out.

An improved version of the term frequency method is the term frequency-inverse document frequency (TF-IDF) method. The intuition is to gauge the uniqueness of a keyword in a document, and hence the relevance. We decide upon a keyword after looking at the fact that this word should be frequent in the current document but not so frequent in the rest of the documents of the given collection. The TF-IDF value for a given keyword $w$ is given as:

$$f_{TF-IDF} = \frac{f_w}{f_{w_{max}}} \log \frac{D}{\rho_w},$$

where, $f_w$ is the frequency of the keyword $w$ in the document, $f_{w_{max}}$ is the maximum frequency of any word in the document, $\rho_w$ is the number of documents in which this keyword occurs, and $\rho$ is the total number of documents in the collection. An alternative formulation is given as:

$$f_{TF-IDF} = \sqrt{\sum_{all\;docs} \left[ \frac{0.5 + 0.5 \frac{f_w}{f_{w_{max}}} \log \frac{D}{\rho_w}}{\left[ \sqrt{\sum_{all\;docs} \left[ 0.5 + 0.5 \frac{f_{w_{max}}}{f_{w_{max}}} \log \frac{D}{\rho_w} \right]^2}} \right]^2}}.$$
Some miscellaneous heuristics may include having some enhanced weightage for the terms occurring in the title, abstract, meta-data or section headings, etc. An alternative motivation is that the authors generally introduce the keywords earlier in a document. Thus, if a word \( w_k \) has \( m \) occurrences as the \( l^{th}, l^{th}, \ldots, l^{th} \) word in the given document, the weight associated with it may be found as:

\[
\text{weight}(w_k) = \frac{\sum_{j=1}^{m} \frac{1}{2^j}}{\sum_{j=1}^{\text{num_words}} \frac{1}{2^j}}.
\]

where, \( \text{num_words} \) indicates the total number of words in the given document. This way the words occurring in the title, abstract or the introduction would get a higher weight, and the frequency of the word would also be taken into account. Now sorting the words on the basis of their weights and taking some top \( N \) words as the keywords is expected to give good result.

2.3 Vector Space Model

An \( n \)-dimensional vector space is taken with one dimension for each possible word or term. Therefore, \( n \) would be the number of words in a natural language. Each document or query is represented as a term vector in this vector space. Each component of the term vector is either 0 or 1, depending on whether the term corresponding to that axis is absent from or present in the text. Alternately, each component of the term vector is taken as a function that increases with the frequency of the term (corresponding to that axis) within the document and decreases with the number of documents in the collection that contain this term. This is to take into account the TF-IDF factor. In such a case, the term vectors of documents must be normalized to one so as to compensate for different document lengths.

Once the term vectors are obtained, the similarity between a document and a query may be obtained by computing the dot product of their term vectors. The larger the dot product, the greater the similarity. We can also calculate the angles between the term vectors of the query and the documents. Then, in response to the given query, the documents may be returned in an increasing order of the angles.

2.4 Boolean Similarity Measures

In literature we also find the Boolean Similarity Measure\(^9, 10\) for computing the similarities of one document to another and documents to queries for automatic classification, clustering and indexing. Such well-known similarity measures are Dice’s Coefficient, Jaccard’s Coefficient, Cosine Coefficient, and Overlap Coefficient. Radecki proposed two similarity measures, \( S \) and \( S’ \), based on Jaccard’s Coefficient. We denote \( \varphi(Q) \) and \( \varphi(R) \) as the sets of documents in the response to query \( Q \) and in the cluster of documents represented by \( R \). The similarity value \( S \) between \( Q \) and \( R \) is defined as the ratio of the number of common documents to the total number of documents in \( \varphi(Q) \) and \( \varphi(R) \).

\[
S(Q, R) = \frac{|\varphi(Q) \cap \varphi(R)|}{|\varphi(Q) \cup \varphi(R)|}.
\]

But, since \( \varphi(Q) \subseteq \varphi(R) \), therefore,

\[
S(Q, R) = \frac{|\varphi(Q)|}{|\varphi(R)|}.
\]

In another measure \( S’ \), Boolean expression \( Q \) is transformed into its reduced disjunctive normal form (RDNF), denoted as \( \widetilde{Q} \), which is the disjunction of a list of reduced atomic descriptors. If set \( T \) is the union of all the descriptors that appear in the to-be-compared Boolean pair, then the reduced atomic descriptor is defined as the conjunction of all the elements in \( T \) in either their original or negated forms. Let \( T_Q \) and \( T_R \) be the sets of the descriptors that appear in \( Q \) and \( R \), respectively. Suppose \( T_Q \cup T_R = \{t_1, t_2, \ldots, t_k\} \), where \( k \) is the set size of \( T_Q \cup T_R \). Then the RDNFs of \( Q \) and \( R \) are:

\[
\widetilde{Q}(T_Q \cup T_R) = (\tilde{q}_{1,1} \land \tilde{q}_{1,2} \land \ldots \land \tilde{q}_{1,k}) \lor\ldots\lor (\tilde{q}_{m,1} \land \tilde{q}_{m,2} \land \ldots \land \tilde{q}_{m,k}),
\]

and

\[
\widetilde{R}(T_Q \cup T_R) = (\bar{r}_{1,1} \land \bar{r}_{1,2} \land \ldots \land \bar{r}_{1,k}) \lor\ldots\lor (\bar{r}_{k,1} \land \bar{r}_{k,2} \land \ldots \land \bar{r}_{k,k}) \lor \bar{r} \land \bar{r} \land \ldots \land \bar{r}.
\]
where \( m \) and \( n \) are the numbers of reduced atomic descriptors in \((Q)_{Q \cup R} \) and \((R)_{Q \cup R} \) respectively, and 
\[
\tilde{q}_{i,j} = \begin{cases} t_j, \text{original} & 1 \leq i \leq m, 1 \leq j \leq k, \\
-t_j, \text{negated} & 1 \leq i \leq m, 1 \leq j \leq k, 
\end{cases}
\]
\[
\tilde{r}_{i,j} = \begin{cases} t_j, \text{original} & 1 \leq i \leq n, 1 \leq j \leq k, \\
-t_j, \text{negated} & 1 \leq i \leq n, 1 \leq j \leq k, 
\end{cases}
\]

where \( \neg \) is the not operator. The similarity value \( S' \) between two Boolean expressions \((Q) \) and \((R) \) is defined as the ratio of the number of common reduced atomic descriptors in \( Q \) and \( R \) to the total number of reduced atomic descriptors in them:
\[
S'(Q, R) = \frac{|(Q)_{Q \cup R} \cap (R)_{Q \cup R}|}{|Q|_{Q \cup R} \cup |R|_{Q \cup R}}.
\]

A new similarity measure \( S^\circ \), with lesser computation, is also proposed, by transforming Boolean expression \( Q \) to its compact disjunctive normal form (CDNF), denoted as \( \hat{Q} \), which is the disjunction of a list of compact atomic descriptors. Each compact atomic descriptor is the conjunction of a subset of descriptors that appear in its own Boolean expression. The CDNFs of \( Q \) and \( R \) are
\[
\hat{Q} = (\hat{q}_{1,1} \land \hat{q}_{1,2} \land \ldots \land \hat{q}_{1,x_1}) \lor \\
(\hat{q}_{2,1} \land \hat{q}_{2,2} \land \ldots \land \hat{q}_{2,x_2}) \lor \ldots \lor \\
(\hat{q}_{m,1} \land \hat{q}_{m,2} \land \ldots \land \hat{q}_{m,x_m}),
\]
and
\[
\hat{R} = (\hat{r}_{1,1} \land \hat{r}_{1,2} \land \ldots \land \hat{r}_{1,y_1}) \lor \\
(\hat{r}_{2,1} \land \hat{r}_{2,2} \land \ldots \land \hat{r}_{2,y_2}) \lor \ldots \lor \\
(\hat{r}_{n,1} \land \hat{r}_{n,2} \land \ldots \land \hat{r}_{n,y_n}),
\]

where \( m \) and \( n \) are the numbers of compact atomic descriptors in \( \hat{Q} \) and \( \hat{R} \), \( x_i \) is the number of descriptors in the \( i^{th} \) \((1 \leq i \leq m)\) compact atomic descriptor of \( \hat{Q} \), and \( y_j \) is the number of descriptors in the \( j^{th} \) \((1 \leq j \leq n)\) compact atomic descriptor of \( \hat{R} \). Each \( \hat{q}_{i,u} \) and \( \hat{r}_{j,v} \) in the CDNFs represents a descriptor in \( T_Q \) and \( T_R \) respectively. Specifically,
\[
\hat{q}_{i,u} \in T_Q, \text{ where } (i,u) = (1...m,1...x_i), \text{ and}
\]
\[
\hat{r}_{j,v} \in T_R, \text{ where } (j,v) = (1...n,1...y_j).
\]

The individual similarity measure is defined as:
\[
s^\circ (\hat{Q}^i, \hat{R}^j) = \begin{cases} 0 \ldots & \text{if } T_Q^i \cap T_R^j = 0 \text{ or } \exists t \in T_Q^i, \neg t \in T_R^j \\
\frac{1}{2^{1/2} - 1} & \text{otherwise}
\end{cases}
\]

where \( \hat{Q}^i \) indicates the \( i^{th} \) compact atomic descriptor of CDNF \( \hat{Q} \), \( \hat{R}^j \) indicates the \( j^{th} \) compact atomic descriptor of CDNF \( \hat{R} \). \( T_Q^i \) and \( T_R^j \) are the sets of descriptors in \( \hat{Q}^i \) and \( \hat{R}^j \) respectively. We define the similarity of two Boolean expressions as the average value of the individual similarity measures \( s^\circ \) between each compact atomic descriptor as:
\[
S^\circ (Q, R) = \frac{\sum_{i=1}^{\hat{Q}} \sum_{j=1}^{\hat{R}} s^\circ (\hat{Q}^i, \hat{R}^j)}{|\hat{Q}| \times |\hat{R}|}
\]

This later method reduces time and space complexity from exponential to polynomial in the number of Boolean terms. We have proposed some improvements in the above mentioned similarity measure \( S^\circ (Q, R) \), as explained subsequently

### 2.5 Enhanced Boolean Similarity Measure

Consider the server descriptors \( R_1 = t_1 \land t_2 \) and \( R_2 = t_1 \land t_2 \land t_3 \), and query \( Q = t_1 \). While \( R_1 \) and \( R_2 \)
are both equally relevant to $Q$, unless $Q$ contains a term with $(-1)$, the similarity measure $S^\circ(Q,R)$ will assign a lower similarity to $R$. In order to overcome this drawback, we drop the terms in the individual similarity measure $s^\circ$, and rewrite it as

$$s^\circ(Q_i, \hat{R}_j) = \begin{cases} 0, & \text{if } |T'_Q \cap T'_R| = 0 \text{ or } \exists t \in T'_Q, t \notin T'_R \\ \frac{1}{2^{|T'_Q - T'_R|}} & \text{otherwise} \end{cases} \quad \text{(3)}$$

In Eq. (2), all the individual similarity measures are added to calculate the overall similarity measure. Adding all the individual similarity measures is not justified due to the following reasoning. Recall that in the basic set theory, $n(A \cup B) = n(A) + n(B) - n(A \cap B)$. However, the coefficient $S^\circ$ attempts to approximate $n(A \cup B)$ by just $n(A) + n(B)$. In order to overcome this drawback, the overall similarity measure is calculated as:

$$S^\circ(Q,R) = \sum_{i=1}^{|Q|} \sum_{j=1}^{|R|} \left[ s^\circ(Q_i, \hat{R}_j) - s^\circ(Q_i, \hat{R}_j) \right]$$

$$\text{(4)}$$

Here, $\hat{R}_j$ is defined as the conjunction of every term appearing in $U_j = T(R_1) \cup T(R_2) \cup \ldots \cup T(R_{j-1})$. The terms in $U_j$ are all those that have already appeared in the first $j-1$ terms of the descriptor $R$. The intuition behind Eq. (4) is to discount the weights for those terms in $R$ which have already been accounted for. This point is further illustrated in Example 1. We can expect the new similarity measure to have the same complexity as LDM, but perform better. Example 1 and 2 illustrate the flaws in the similarity measure $S^\circ$, and our correction to these flaws.

Example 1

This example illustrates the first flaw in $S^\circ$, and how $S^\circ$ can overcome the drawback. Consider a server $R = (t_1 \land t_2) \lor (t_1 \land t_3) \lor (t_1 \land t_4) \lor \ldots \lor (t_1 \land t_k)$ and a query $Q = (t_1 \land t_2)$. In other words, we have chosen the server $R$ such that the term $t_1$ appears in all its terms. This will lead to a similarity value

$$S^\circ = \frac{1}{2^1} + \sum_{j=2}^{(k-2)} \frac{1}{2^j} = \frac{1}{2} + \sum_{j=2}^{(k-2)} 0 = \frac{1}{2}.$$

In other words, $S^\circ$ assumes a high value, proportional to the number of terms in the query, even though the server is not highly relevant to query. (The term $t_2$ does not appear in the server $R$ at all.) If we use $S^\circ$ to calculate the similarity, we get

$$S^\circ = \frac{1}{2} + \sum_{j=2}^{(k-2)} \frac{1}{2^j} = \frac{1}{2} + \sum_{j=2}^{(k-2)} 0 = \frac{1}{2}.$$

which is a constant, irrespective of the number of times the term $t_1$ is repeated in the server. Note that the similarity value is also not very high.

Example 2

This example illustrates the other flaw in $S^\circ$. Consider a query $Q = t_1$, and the servers

$R_1 = t_1 \land t_2$, \hspace{1cm} R_2 = t_1 \land t_2 \land t_3$,

$R_3 = t_1 \land t_2 \land t_3 \land t_4$, \ldots ,

$R_k = t_1 \land t_2 \land t_3 \land \ldots \land t_{k+1}$

Then, using Eqs (1) and (2) the similarity value for the server $R_i$ for the query will be

$$S^\circ(Q,R_i) = \frac{1}{2^0 + 2^1 - 1} = \frac{1}{2^1}.$$

In other words, different servers are assigned different similarity values, even though they are all equally relevant. Using our similarity value, we find that $S^\circ(Q,R_i) = \frac{1}{2^0} = 1$ for all the servers.

Although $S^\circ$ overcomes two of the drawbacks in the $S^\circ$, it does not take into account the number of relevant documents. As a result, it cannot come close
We now propose a second similarity measure $S^{9}$, which takes into account the number of relevant documents on the server as well as the total number of documents on the server. We associate with each term in server descriptor, a weight that signifies the importance of the term in each document of the server. The weights are in the range 0..1, and add up to 1.0. An example of such a server descriptor is $SD = (t_{1} \land t_{2})^{0.6} \lor (t_{3} \land t_{4})^{0.4}$. Here, it means that 60 per cent of the server documents contain $(t_{1} \land t_{2})$ and 40 per cent of the server documents contain $(t_{3} \land t_{4})$. In general, a server descriptor is of the form $SD = (S_{1}, w_{1}) \lor (S_{2}, w_{2}) \lor \ldots \lor (S_{k}, w_{k})$, where $S_{i}$ represents AND terms and $w_{i}$ are weights with them, such that $\sum_{i=1}^{k} w_{i} = 1$. The new individual similarity measure is defined as $S^{9}(Q_{i},R_{j}) = w_{j} \cdot S^{9}(Q_{i},R_{j})$. Thus,

$$
S^{9}(Q_{i},R_{j}) = \begin{cases} 
0 \ldots \text{if } T_{Q_{i}} \land T_{R_{j}} = 0 \lor \forall t \in T_{Q_{i}}, \exists t \in T_{R_{j}}
\end{cases}
$$

The overall similarity measure will be:

$$
S^{9}(Q,R) = \sum_{i=1}^{\|Q\|} \sum_{j=1}^{\|R\|} \left[ S^{9}(Q_{i},R_{j}) - S^{9}(Q\setminus Q_{i},R\setminus R_{j}) \right].
$$

These measures are benchmarked against the standard CISI dataset. It is shown that it can be said with 64 per cent confidence that similarity measure $S^{9}$ performs better than the measure $S^{8}$, and with 99.75 per cent confidence that similarity measure $S^{9}$ performs better than the measure $S^{8}$.

### 2.6 Problems with Content Based Web Search

The main problem of the content-based web searching stems from what is called as "search engine persuasion" or "keyword spamming" or simply "spamming". The commercial interest of many web page authors to have their pages ranking high for certain queries forces them to do some illegitimate manipulations. These manipulations may go as far as adding text in an invisible ink. For instance, if the word "car" is repeated some thousand times in an invisible link, to the web surfer it would not appear awkward, but the search engine would rate it very high for the queries on "car". The power of hyperlink analysis, as we discuss in the subsequent sections, comes from the fact that it uses the content of other pages to rank the current page. This gives an unbiased factor to the ranking.

There may also be cases where the keyword we are looking for may not at all appear in a page, which, otherwise, is very much relevant to the query. For instance, the home page of IBM did not contain the word "computer". That would mean that if we rely solely on the content-based techniques, the pointer to the home page of IBM would not at all be returned in response to a query on "computer". This is very clearly an undesirable situation. We shall see how these problems are taken care of in the following sections.

### 3 Connectivity-Based Web Search

In an Internet search, the user uses a query language to describe the nature of documents, and, in response, a search engine returns a ranked list of documents that "best match" the description. It is this document ranking, which is felt to be most crucial. The hyperlink structure of the web has been found to help to a great extent in this respect. We shall, therefore, carry out a review of the existing techniques for hyperlink based ranking.

The $HyPursuit^{15}$ prototype is a scalable system that uses content link hypertext clustering based on document contents as well as the link information, to structure the information space and to support the entire range of search activities. The similarity measure between the two web documents based on the links between them includes the measures of:

(a) Shortest path between the two documents,

(b) The number of ancestor documents that refer to both documents in question, and

(c) The number of descendent documents that both refer to.

The web is modeled as a directed graph, with the web pages being the nodes and the hyperlinks between them being the edges of the model graph. One of the earliest efforts for harnessing the
The hyperlink structure of the web is reported by Kleinberg. Therein, we come across, what is called the Hyperlink-Induced Topic Search (HITS) algorithm. Beginning with a set of seed pages obtained from a term-based search engine (such as AltaVista) pertaining to the topic, a base sub-graph is obtained by including in the set all those pages that are pointed to by the seed pages, as also all those which point to the seed pages. This sub-graph is then subjected to a weight propagation algorithm, so as to determine the numerical estimates of hub and authority weights by an iterative procedure. HITS, then returns as hubs and authorities for the search topic those pages with the highest weights.

Clever realizes the concepts of Authorities (pages which provide the best source of information on a given topic) and Hubs (pages which provide collection of links to authorities). We associate a non-negative authority weight \( y_p \) and a non-negative hub weight \( x_p \) with each page. We update the value of \( x_p \) for a page \( p \), to be the sum of \( y_q \) over all pages \( q \) that link to \( p \):

\[
x_p = \sum_{q \text{ such that } q \rightarrow p} y_q
\]

where the notation \( q \rightarrow p \) indicates that \( q \) links to \( p \). In a strictly dual fashion, if a page points to many good authorities, we increase its hub weight via

\[
y_p = \sum_{q \text{ such that } p \rightarrow q} x_q
\]

Let us number the pages \( \{1, 2, ..., n\} \) and define their adjacency matrix \( A \). Let us also write the set of all \( x \) values as a vector \( x=(x_1, x_2, ..., x_n) \), and similarly define \( y=(y_1, y_2, ..., y_n) \).

Then the update rule for \( x \) can be written as

\[
x \leftarrow A^T y,
\]

and the update rule for \( y \) can be written as

\[
y \leftarrow A x.
\]

Unwinding these one step further, we have:

\[
x \leftarrow A^T y \leftarrow A^T A x \leftarrow (A^T A)x, \quad \text{and}
\]

\[
y \leftarrow A x \leftarrow A A^T y \leftarrow (A A^T)y.
\]

Linear Algebra tells us that, when normalized, this sequence of iterates converges to the principal eigenvector of \( A^T A \) for \( x \) and \( A A^T \) for vector \( y \).

Google adopts the concept of PageRank. The quality measure of a page is its indegree. Suppose there are \( T \) total pages on the web. We choose a parameter \( d \) such that \( 0 < d < 1 \), a typical value of \( d \) might lie in the range \( 0.1 \leq d \leq 0.15 \). Let pages \( p_1, p_2, ..., p_m \) link to page \( p_i \), and let \( C(p_i) \) be the number of links out of \( p_i \). Then the PageRank \( R(p_i) \) of the page \( p_i \) is defined to satisfy Eq. (5).

\[
R(p_i) = \frac{d}{T} + (1-d) \sum_{i=1}^{m} \frac{R(p_i)}{C(p_i)}, \quad \ldots \quad (5)
\]

Note that the PageRanks form a probability distribution over web pages, so that the sum of all web pages' PageRank will be one. PageRank can be calculated using a simple iterative algorithm, that corresponds to the principal eigenvector of the normalized link matrix of the web. PageRank can be thought of as a model of user behavior. We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page. The probability that the "random surfer" visits a page is its PageRank. The \( d \) damping factor is the probability at each page that the "random surfer" will get bored and request another page. A page can have a high PageRank if there are many pages that point to it, or alternatively, if there are some pages with high PageRanks that point to it.

A note is in order here. Apparently it seems that Clever is more versatile than Google, but still Google is found to give reasonably good results. The prime reason behind this is that a good hub would become a good authority in due course of time. The analogy may be taken from the referencing of research papers. An original research paper is a good authority and is often cited by other papers, while a good survey paper is a good hub as it contains pointers to some good original research works. But we see that in due course of time, a survey paper also gets cited very frequently and so qualifies to become as good an authority as any other research paper. This is the reason why Google gets away with just the notion of PageRank, without separating the concepts of authorities and hubs.
Some improvements in calculating the hub and authority weights are suggested\(^9\). Instead of summing up the authority weights to get the hub weight of a page that links to all those authorities, it is reasoned out to do the averaging. This way, a hub would be better only if it links to good authorities, rather than linking to both good and bad authorities. Moreover, a hub-threshold as well as an authority-threshold is also suggested. While computing the authority weight, only those hubs should be counted whose hub weight lies above the current average hub weight. This means that a site should not be considered a good authority simply because a lot of poor hubs point to it. Similarly, while computing the hub weight, only some top \(K\) authorities should be counted. This amounts to saying that to be considered a good hub, the site must point to some of the best authorities.

**A Stochastic Approach for Link-Structure Analysis (SALSA)**\(^{10}\) is also an iterative algorithm, which starts with a similarity-constructed base set \(C\) (as in Kleinberg\(^5\)). The approach is based upon the theory of Markov chains, and relies on the stochastic properties of random walks performed on the base set \(C\). It differs from the earlier work\(^3\) in the manner in which the association matrices are defined. It performs a random walk by alternately going uniformly to one of the pages (a) which links to the current page and (b) linked to by the current page. The authority weights are defined to be the stationary distribution of the two-step chain doing first step (a) and then (b), while hub weights differ in doing step (b) first and then (a). If \(B(i) = \{k: k \rightarrow i\}\) denotes the set of all nodes that could be reached from \(i\) by following a back link, and \(F(i) = \{k: i \rightarrow k\}\) denotes the set of all nodes that could be reached from \(i\) by following a forward link, then the transition probabilities of the Markov chains for the authorities may be given as:

\[
P_a(i, j) = \sum_{k \in B(i) \cap F(j)} \frac{1}{|B(i)||F(i)|}.
\]

For this Markov chain, the stationary distribution \(a = \{a_1, a_2, \ldots, a_N\}\) gives the required authority weights for a total of \(N\) pages on the web. Similarly, the transition probabilities of the Markov chains for the hubs may be given as:

\[
P_h(i, j) = \sum_{k \in F(i) \cap B(j)} \frac{1}{|B(i)||F(i)|}.
\]

For this Markov chain, the stationary distribution \(h = \{h_1, h_2, \ldots, h_N\}\) gives the required authority weights for a total of \(N\) pages on the web.

A Markov Model based rank estimation technique has been presented by Zhang and Dong.\(^{11}\) The advantage with this approach is that it is no more an iterative one, rather it just requires solving a set of simultaneous equations. Suppose for a query \(q\), the set of related web resources returned by the search engine are \(R = \{r_1, r_2, \ldots, r_n\}\), then this \(R\) may be taken as the state space for the Markov Model. If at time \(t\), the virtual user is browsing a web resource \(r_t\), then at \((t+1)\), he may do one of the following:

(i) Continue browsing the resource \(r_t\); this is said to be the relevance parameter \((\alpha)\).

(ii) Click a hyperlink in \(r_t\) to jump to a new web resource; this is authority parameter \((\beta)\).

(iii) Press the "Back" button of the browser; this is integrativity parameter \((\gamma)\).

(iv) Select another web resource from \(R\); this is denoted as the novelty parameter \((\epsilon)\).

These parameters are chosen such that \(\alpha + \beta + \gamma + \epsilon = 1\). These parameters would vary for different users. But they have been initially set as \(\alpha = 0.6, \beta = 0.2, \gamma = 0.19\) and \(\epsilon = 0.01\). For the directed linkage structure graph \(G = (V, E)\), the tendency matrix for the set of related web resources \(R\) is:

\[
U = (u_{i,j})_{n \times n},
\]

where

\[
u_{i,j} = \begin{cases} 
\alpha \times \text{sim}(r_i, q), & \text{if } i = j; \ i.e. \ for \ all \ diagonal \ elements \\
\beta, & \text{if } (v_i, v_j) \in E; \ i.e. \ when \ there \ is \ a \ hyperlink \ from \ v_i \ to \ v_j \\
\gamma, & \text{if } (v_i, v_j) \in E; \ i.e. \ when \ there \ is \ a \ hyperlink \ from \ v_j \ to \ v_i \\
\epsilon, & \text{otherwise}
\end{cases}
\]

Here, \(\text{sim}(r_i, q)\), the relevance function lying between \(0\) and \(1\), depicts the similarity between the query \(q\) and the web resource \(r_i\). This relevance may be calculated based on the conventional Information Retrieval techniques, such as Term Frequency (TF), Inverse Document Frequency (IDF), word weight, meta-data, etc. Now the transition probability matrix for \(R\) may be found as:

\[
\begin{align*}
P_h(i, j) &= \sum_{k \in F(i) \cap B(j)} \frac{1}{|B(i)||F(i)|} \\
\end{align*}
\]
The main advantage of Eqn. (6) over Eqn. (5) lies in the fact that getting to know the back links of a page is not as easy and straightforward as its forward links. The back links of a page may be found only after the whole web has been thoroughly crawled. With the current huge size of the web, as also the growth of the web size being exponential, this intensive exercise of crawling the whole web, may take hours or even days to finish. On the other hand, the forward links of a page are known the moment the contents of only that very page have been retrieved. So with Eqn. (6), the computation of the PageRank may start the moment we retrieve a page, and we don’t have to wait for the crawl of the whole web to be completed, as is the case with Eqn. (5).

However, we see a serious disadvantage with Eqn. (6). Since we are dependent on the contents of the current page, the problem of spamming would be back. The web page authors may now provide links to some good pages in an invisible link or even otherwise, so that the PageRanks of their pages gets boosted. To do away with this problem, we propose a hybrid technique, which takes the effect of both the forward as well as the back links. For this, we choose another parameter \( \alpha \) (\( 0 < \alpha < 1 \)), in addition to \( d \), such that \( 0 < (d + \alpha) < 1 \). Now if a page \( P_i \) is pointed to by pages \( P_1, P_2, \ldots, P_m \), with \( n(P_i) \) being the number of links out of \( P_i \), and points to pages \( P_1, P_2, \ldots, P_m \), with \( D(P_k) \) being the number of links incoming to \( P_k \), then the PageRank \( R(P_i) \) of the page \( P_i \) may be defined as:

\[
R(P_i) = \frac{d}{T} + (1 - d) \sum_{j=1}^{m} \frac{R(P_j)}{D(P_j)} + \alpha \sum_{k=1}^{n} \frac{R(P_i)}{C(P_k)} \quad \text{... (7)}
\]

Let us now see what advantages do we gain out of the modifications resulting in Eqn. (7). Firstly, since we have considered both the forward as well as the back links simultaneously, we may expect faster convergence. Secondly, we can restrict the problem of spamming by choosing an appropriate value of \( \alpha \). Lesser the value of \( \alpha \), lesser the weightage of the contents of the current page, and so lesser the chances of spamming. The third advantage of Eqn. (7) may be grasped when we consider the intuition behind these formulations. With Eqn. (5), PageRank can be thought of as a model of user behavior. We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting

\[
P = (p_{ij})_{n \times n}, \text{ where } p_{ij} = \frac{u_{ij}}{\sum_{j=1}^{n} u_{ij}}
\]

It is claimed that the ultimate (stable) distribution vector \( \pi = (\pi_1, \pi_2, \ldots, \pi_n) \) is the unique solution of the equation

\[
\pi P = \pi, \text{ that satisfies } \pi_i > 0 \text{ and } \sum_{i=1}^{n} \pi_i = 1.
\]

This way, given a set of related web resource \( R \), we can construct the transition probability matrix \( P \) using the above equations. Then the ultimate (stable) distribution vector \( \pi \) may be found by solving the thus obtained set of equations. This vector \( \pi \) is just the rank we wanted to estimate for all the pages in \( R \).

A machine learning perspective of connectivity based web search is given by Chakrabarti. The application of data mining and machine learning procedures for analyzing the hypertextual structure of the web is surveyed therein. The algorithms from all three domains, namely, supervised, semi-supervised and unsupervised, are discussed vis-à-vis hyperlinks.

3.1 Integrating Hub-Authority and PageRank

We understand that the PageRank of a web page must integrate the hub and authority weights of that page. But the Eq. (5), which is used for the PageRank calculation, considers only the back links of a page. This means that the PageRank calculation is biased more towards the authority weight of that page and neglects its hub value altogether. This thought leads us to the dual of Eq. (5)\(^2\). The iterative algorithm would converge to the correct value even when we consider the forward links, instead of the backward links. If a page \( P_i \) points to pages \( P_1, P_2, \ldots, P_m \), and let \( D(P_i) \) be the number of links incoming to \( P_i \), then the PageRank \( R(P_i) \) of the page \( P_i \) may be defined as:

\[
R(P_i) = \frac{d}{T} + (1 - d) \sum_{j=1}^{m} \frac{R(P_j)}{D(P_j)} \quad \text{... (6)}
\]

We expect Eqn. (6) to perform as good as Eqn. (5). It is shown\(^2\) that the PageRank calculation using Eqn. (6) still qualifies to form a probability distribution over web pages.
"back" but eventually gets bored and starts on another random page. The probability that the "random surfer" visits a page is its PageRank \( R(p_i) \). The \( d \) factor is the probability at each page that the "random surfer" will get bored and request another page. But strictly speaking, this is not too realistic. Just think how often do we make use of the "back" button of the browser. With Eqn. (7), we think we have removed this intuitive problem. Now with probability \( R(p_j) \), the user visits the page \( p_j \), with probability \( d \), he requests a page at random, and with probability \( \alpha \), he hits the "back" button. So this way, we think that our formulation is closer to reality. The fourth advantage comes from the basic fact that now none of the two basic notions, namely the hubs and authorities are being neglected while estimating the PageRank.

One may argue that, although the simultaneous consideration of both the forward as well as the back links may give faster convergence, the overall time required for the calculation of PageRanks may be more than what is required while using Eqn. (6). The reason for this may be given to be the non-availability of the back links until and unless the crawl of the whole web is completed. We have thought of a way out of this problem. We begin the PageRank calculation the moment the contents of the current page are available, with the information about the forward links coming from the current contents of that page, while the information about its back links coming from the just-previous crawl. This way, we would compromise the PageRank precision for the turn around time of the PageRank calculation process.

### 3.2 Integrating Content Based and Connectivity Based Techniques

Of late, there have been efforts to merge the content based and the connectivity based techniques for better effects. For instance, an exclusive study has been made to combine four different content based techniques with the HITS-based algorithms. It has been argued that since the pure connectivity based techniques are affected by the Tight Knit Community (TKC) Effect (see section 4), they may give poor results. This problem is specifically attributed to the cases where the number of in-links to a page is smaller as compared to the number of out links. It is claimed that this problem cannot be solved but by the introduction of content based techniques along with the connectivity based HITS algorithm.

In another case, the PageRank algorithm is shown to be improved substantially with the introduction of what is called as topic sensitivity. In the conventional PageRank algorithm, the PageRank vector is pre-computed independent of any particular search query. For topic sensitive PageRank algorithms, a set of PageRank scores is pre-computed for each page, each with respect to one of various topics. At query time, these PageRank scores are combined based on the topics of the query. The topic of the query may be picked up from either the history of queries, or the user's bookmarks and browsing history, or the document itself in the case of the query term highlighted by the user.

### 4 Finding Related Pages

Since it has been well recognized that the hyperlink structure can be very valuable for locating information on the web, an effort has been made to find related pages on the web by harnessing this hyperlink structure. The motivation for these works comes from the fact that forming good queries can be a difficult task, especially in an information space unfamiliar to the user. Thus, a different approach to web searching is adopted where the input to the search process is not a set of query terms, but instead is the URL of a page, and the output is a set of related web pages.

To this aim, two algorithms namely, the Companion Algorithm and the Cocitation Algorithm are proposed. The companion algorithm is derived from the HITS algorithm, which essentially uses the concept of authorities and hubs, as discussed earlier. The precise extension is that it not only exploits the links but also their order on a page. The cocitation algorithm finds pages that are frequently co-cited with the input URL \( u \), i.e., it finds other pages that are pointed to by many other pages that all also point to \( u \). When compared to the Netscape's 'What is Related' service (http://home.netscape.com/escapes/related/), it is found that the precision at 10 for these two algorithms are 73 per cent and 51 per cent better than that of Netscape, despite the fact that Netscape uses both content and usage pattern information in addition to connectivity information.

A work somewhat similar in the goal but different in approach is reported. A new hypertext resource discovery system called a Focused Crawler is described therein. The goal is to selectively seek out pages that are relevant to a predefined set of
topics, which are specified not using keywords but using exemplary documents. Rather than collecting and indexing all accessible web documents to be able to answer all possible ad-hoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the web. This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date.

A similar effort has also been made in the Shark-Search Algorithm\(^5\). The crawl is encouraged in that direction where the relevance is found to be increasing or at least sustaining. The diminishing relevance in a particular direction discourages the crawl and so the crawling is kept focussed in a particular domain only. This is a dynamic web search algorithm in the sense that it does not rely on the previously built indices, rather fetches the data at the time the query is issued.

The hyperlink structure along with the text surrounding the hyperlink is utilized\(^*\) for finding the related pages. The technique is named as Automatic Resource Compilation (ARC). The algorithm has three distinct phases: a search-and-growth phase, a weighting phase, and an iteration-and-reporting phase. The first phase is similar to the base set formulation as in Kleinberg\(^6\). The expansion step, in which all those pages are included that either link to or are linked by any of the pages in the root set, is successively carried out twice. The broader set thus obtained is called the augmented set. In the weighting phase, not only do we estimate the authority weights and the hub weights as usual, but also we bring in the effect of the text surrounding the hyperlink. This is done with a faith that a hyperlink is usually preceded or succeeded by its description, and hence the utility with our aim of discovering related pages. It has been experimentally found that a text window of size 50 bytes, called the anchor window, on either side of the hyperlink, is sufficient to capture this intuition. To the authority weights of each page, pertaining to the topic in discussion, we add the number of occurrences of that topic term in the anchor window. We do, however, normalize after each iteration to keep the entries small. We must iterate to get a unique steady state value of the authority and hub weights. This convergence was found to be pretty fast, primarily because the interest was to stabilize the weights of just top few, say 15, pages. This type of "near convergence" required just around 5 iterations.

Hence, after 5 iterations, the top 15 pages were reported to be the set of pages related to the given topic.

In all the above approaches for finding related pages, or the web communities, a serious drawback has been observed in a few instances of experimentation. The phenomenon has been termed as "topic drift"\(^9\). We do get what are called the "tightly knit communities" (TKC) with the above approaches, but these TKCs may not be relevant to the topic we began our processing with. For instance, with the topic being "jaguar", the HITS algorithms does converge to give a TKC, but it happens to be all about the city of Cincinnati. This may be due to the fact that a large number of online newspaper articles in the Cincinnati Enquirer discussing the Jaguars Football team, also link to the same standard Cincinnati Enquirer service pages. Interestingly, with the query term "abortion" also, the HITS algorithm converges to the TKC containing the pages about the city of Cincinnati. Research efforts are on to counter this problem of topic drift, which gets serious in some occasional instances.

An automated procedure for evaluating the systems that finds related pages has already been outlined\(^6\). For this, a familial distance is defined between two documents as the distance between the most specific classes of the two documents in the class hierarchies on the web, such as Open Directory. Based on the familial distance between the sample document and each of the result documents, an ordering of the result documents may be carried out. This ordering of the documents is compared with the ordering returned by the system of finding related pages.

5 Conclusion

We have briefly reviewed the classical content-based information retrieval techniques, and the problems arising out of it when applied to the World Wide Web. We have seen how the hyperlink structure of the web is harnessed for overcoming these problems. The notion of PageRank appears to be central to the solution of the web search problem. The PageRank of a web page gives the relative importance of that page in comparison to the other pages found on the web. The very name PageRank suggests that it could directly be used for ranking the results to a search query. For finding the neighborhood of a page and hence the tightly knit
communities (TKC), the hyperlinks are directly used in some approaches. There have also been approaches in which PageRank has been used as a guideline to seek first/only the more important related pages. As a bottom line, we can say that the phenomenal success of Google search engine speaks volumes of the successful paradigm shift from the content-based to a connectivity-based search for the World Wide Web.

However, the fact remains that the schemes for calculating PageRank of a page are biased towards the authority of the page and neglects the hub value of that page. There is a need to improve the calculation of PageRank by not only trying to remove this bias, but also trying to bring down the computational complexity involved. While finding web communities, there is a need to come up with techniques that could do away with what has been described as the "topic drift" problem. We need to have techniques for quantifying the quality of search results of the search engines. All these may be taken as the research issues emerging out of the discussions in this paper.

Acknowledgement

Thanks are due to the anonymous referee for the valuable suggestions for improving upon the draft of this paper.

References

4 Google, http://www.google.com


