
World Wide Web, 4: 189–207 (2001)


# A Probabilistic Approach for Distillation and Ranking of Web Pages *

GIANLUIGI GRECO, SERGIO GRECO and ESTER ZUMPANO {ggreco;greco;zumpano}@si.deis.unical.it
*DEIS, Università della Calabria, 87030 Rende, Italy*

*Abstract*

A great number of recent papers have investigated the possibility of introducing more effective and efficient algorithms for search engines. In traditional search engines the resulting ranking is carried out using textual information only and, as showed by several works, they are not very useful for extracting relevant information. Present research, instead, takes a new approach, called *Topic Distillation*, whose main task is finding relevant documents using a different similarity criterion: retrieved documents are those related to the query topic, but which do not necessarily contain the query string. Current algorithms for topic distillation first compute a base set containing all the relevant pages and then, by applying an iterative procedure, obtain the *authoritative* pages. In this paper, we present a different approach which computes the authoritative pages by analyzing the structure of the base set. The technique applies a statistical approach to the co-citation matrix (of the base set) to find the most co-cited pages and combines a link analysis approach with the content page evaluation. Several experiments have shown the validity of our approach.

**Keywords:** information retrieval on the Web, Web searching, search engines, random walks

## 1. Introduction

The increasing popularity of the Web has significantly broaden the possibility of sharing ideas and human knowledge on a scale never seen before. Search services on the WWW are becoming increasing popular among users because of the huge amount of data available [3]. Despite its success, as it appears from the results supplied by traditional term-based search engines, retrieving and filtering information on the Web is actually quite a difficult task. In traditional search engines the resulting ranking is carried out by using textual information only and thus they are not very useful for extracting relevant information. Indeed the retrieved result is affected by the users' precision in expressing the query and is strictly related to the query string, i.e., documents are ranked just according to a degree of textual similarity with respect to the user query [10,14,21,23,25,27]. However, usually, users are unclear about the information they need and so they do not give much thought to query formulation. Moreover, if the query pertains to topics which are abundant on the Web, search services become unusable because of the huge number of pages obtained (often millions of pages). For instance, at the time of this work, the search engine AltaVista [13]

* Work partially supported by MURST grants under the projects "Data-X" and "D2I." The second author is also supported by ISI-CNR. A preliminary version of this paper appeared in [18].

returned more than 29,000,000 pages in reply to the query asking for the documents related to the word "java".

Current research, instead, takes a different approach, which goes under the name of *Topic Distillation* on the Web [9,10,21,24]. It basically consists in finding documents related to the query topic, but which do not necessarily contain the query string. The following classical example shows the advantage of this [21]. If we want to find Web pages associated with the query string "search engine," a Term-Based search engine is not useful because it does not return pages like `www.yahoo.com`, `www.altavista.com` or `www.excite.com`. This happens since none of the really interesting pages contains the query string. The purpose of Topic Distillation is to increase the precision of the search algorithm in order to return the most relevant pages, even if there is no trace of the query string in them.

In order to achieve this, Kleinberg [21] observed that there is an additional source of information that can be used for searching the Web: its structure made of nodes (Web pages) connected through arcs (links among pages). Using this idea, he proposed a connectivity analysis algorithm, called *mutual reinforcement* approach. In fact, a link which is not made for a navigational purpose, encapsulates a human judgment on the page relevance with respect to a certain topic. As a consequence, all pages can be divided into two groups: hubs and authorities. An authority is a relevant page pointed to by many hubs; while a hub is a page that points to many relevant ones. Kleinberg's algorithm computes an authority score for each page as an indicator of relevance.

In searching the Web a second important aspect must also be considered: the automatic discovery of communities connected to a given topic, known as *topic enumeration*. Topic search and enumeration are tightly related since in the searching of documents on a given topic it is useful to compute the most authoritative documents, but also to identify the different communities.

In this paper, we present a technique which, by exploiting the graph structure of the Web, improves the quality of both topic search and enumeration. Our technique is based on the application of a statistical approach to the co-citation matrix [28] (associated with the *base set* obtained in the first step of Kleinberg's algorithm or some other algorithm based on the mutual reinforcement approach) to find the most co-cited pages. The different communities are then derived directly from the co-citation matrix (also called similarity-matrix) and the most relevant pages are those which are the "most similar" to all the other pages in the same community. Our technique is more general and efficient than previous techniques proposed in the literature since it is able to identify the different communities and the most authoritative pages without using any iterative procedure. Techniques for topic distillation can be applied to Web data (e.g., HTML data) and, generally, to data which can be modelled by graphs such as XML and semistructured data [1,2,16,22].

To show the validity of our approach, we have developed a system prototype for topic distillation and enumeration of Web documents, called $\mathcal{STED}$. Several experiments have demonstrated the effectiveness and efficiency of our technique.

The rest of this paper is organized as follows. Section 2 contains a descriptions of the current techniques used for page ranking and topic distillation. Section 3 presents our probabilistic approach for topic distillation; this section also contains the theoretical results

useful for its implementation and a comparison of our technique with other approaches. Section 4 describes a prototype developed at the University of Calabria. Section 5 shows some experimental results comparing our algorithm with Kleinberg's algorithm. Finally, Section 6 contains our conclusions.

## 2. Page ranking and authoritative Web pages

This section describes some techniques and algorithms for the ranking of Web pages. These techniques can be divided into two distinct groups: techniques based on a statistical approach for ranking (e.g., the PageRank Algorithm [7,8]) and techniques for topic distillation based on Kleinberg's algorithm [6,21]. Since the two basic approaches have some limitations, several works have proposed extensions of the basic algorithms avoiding some drawbacks in the computation of relevant pages [23]. Before reviewing current techniques for page ranking, let us present the formal definition of the (Web) graph and its representation.

Let $\Gamma$ be a set of node identifiers and $\Sigma$ the alphabet of edge labels. A graph $G$ over $\Gamma$ and $\Sigma$ is a pair $(V, E)$, where $V \subseteq \Gamma$ is the set of nodes and $E \subseteq \{(u, \sigma, v) \mid u, v \in V, \sigma \in \Sigma\}$ is the set of labeled edges. A graph over $\Gamma$ and $\Sigma$ is said to be weighted if $\Sigma = \mathcal{R}^+$ or $\Sigma = \mathcal{Z}$, i.e., the alphabet of edge labels is the set of positive reals or the set of rational numbers.[1]

A weighted graph $G = (V, E)$ is stored by means of a $(|V| \times |V|)$ matrix $A$, called adjacency matrix. The value of $A_{i,j}$ denotes the weight (label) of the arc $i \rightarrow j$ and $A_{i,j} = 0$ means that there is no arc from $i$ to $j$. Let $A$ be a matrix, an *eigenvalue* of $A$ is a number $\lambda$ such that $Aw = \lambda w$, for some $w$ called *eigenvector*. The number of linearly independent eigenvectors defines the multiplicity of the eigenvalue $\lambda$.

### 2.1. PageRank algorithm

Techniques for ranking Web pages have been successfully used in search engines such as *Google* [17]. Google is a system that, for the first time, implements an algorithm which uses the link structure of the Web to calculate a quality ranking for each page: the *PageRank* algorithm. Mathematically the PageRank measure can be expressed by the following formula [8].

*Definition 2.1.* Let $p$ be a Web page, $out(p)$ the set of links starting from $p$. The PageRank of $p$ is:

$$PR(p) = d + (1 - d) \sum_{p_i \in L(p)} \frac{PR(p_i)}{|out(p_i)|},$$

where $d$ is a real number such that $0 < d < 1$.

PageRank simulates the behaviour of a surfer randomly navigating the Web, who jumps with probability $d$ to a page selected randomly and, consequently, he follows a link selected randomly with probability $1 - d$. The value of $d$ is fixed and a typical value is in the range $[0.1,\ 0.15]$. It follows that $PR(p)$ can be calculated by initially assigning the value 1 to every page and then by using a simple iterative algorithm. The retrieved solution corresponds to the principal eigenvector of the normalized link matrix of the Web. Note that $PR(p)$ can be scaled so that $\sum_{\forall p} PR(p) = 1$; in such a case $PR(p)$ can be thought of as a probability distribution over pages and hence a weight function. Therefore, PageRank has a simple interpretation: it gives the probability that a random surfer reaches the given page starting from a random page. Such a probability is a natural candidate for capturing the intuitive notion of page relevance, that is, the value of $PR(p)$ is a measure of the relevance of page $p$.

In any case, it is easy to see that PageRank is just a mechanism for ranking pages and not a system for topic distillation; that is, the rank is calculated a priori, just on the basis of the link structure: the ranking of a page $p$ is not affected by the query string. In some way it encapsulates a concept of generic popularity, but not of relevance for a particular topic. Moreover, we observe that:

- PageRank is based on the hypothesis that from each page the probability of following an outgoing link is the same (as a consequence of not considering the query string).
- It does not make any assumptions about the length of the random-walks, because it only calculates the asymptotical distribution of the random walk.
- It does not consider the possibility that a Web-surfer would remain on a certain page (if it is relevant in itself).

### 2.2. *Kleinberg's algorithm*

Kleinberg proposed an innovative approach based on the observation that each page has an *authority* rating (based on its incoming links) and a *hub* rating (based on its outgoing links). Kleinberg's algorithm, called *mutual reinforcement approach*[2], consists of two phases:

- *Creation of the base set.* A *root* set of documents matching the query is obtained by taking the first $t$ documents given by a traditional search engine. Then, the *root* set is augmented by adding the pages that point to or are pointed to by documents in *root* (for a maximum of $c$ pages). The process can be repeated several times and the resulting set, called *base* set, is used for the successive step. Kleinberg suggests the values $t = 200$ and $c = 50$.
- *Computation of hub and authorities scores.* Let $n$ be the number of pages in the *base* set $\mathcal{B}$, the data structure used by the algorithm is an $n \times n$ adjacency matrix, $A$, where $A_{i,j} = 1$ if there are one or more hyperlinks from page $i$ to page $j$, otherwise $A_{i,j} = 0$. Let $k$ be the iteration number and let $X$ and $Y$ two vectors of size $n$ representing the authority and hub scores for each page in $\mathcal{B}$. The algorithm applies the following steps to the *base* set:

  1. $X = Y = [1, \ldots, 1] \in \mathbb{R}^n$.

2. For all the $k$ iterations needed

   (a) $X' \leftarrow A^{\mathrm{T}}Y$, $Y' \leftarrow AX$,

   (b) $X = \dfrac{X'}{|X'|}$, $\quad Y = \dfrac{Y'}{|Y'|}$.

At the end of the iteration process, the vectors $X$ and $Y$ contain, respectively, the authority and hub scores assigned to all pages in the base set. Kleinberg proved that after a sufficient number of iterations $k$ (generally, not larger than 20), the vectors $X$ and $Y$ converge to the principal eigenvectors of the matrices $A^{\mathrm{T}}A$ and $AA^{\mathrm{T}}$, respectively. The principal eigenvector of the transition matrix corresponds to the largest eigenvalue and identifies the largest "community" of Web pages; consequently, pages outside this community are not considered.

An important technical assumption made for the convergence of Kleinberg's algorithm is that the principal eigenvalue should have unitary multiplicity, so that just one eigenvector is associated to it. Although the convergence is not affected by this assumption, we note that the algorithm can produce unexpected results.

Consider Figure 1 where we report a graph in which this problem is emphasized; we can see the two communities {2,3} and {5,6}, but the algorithm ranks the nodes 2, 3, 5 and 6 at the same position[3] causing a user to think that these nodes belong to the same community. Instead, what a user would actually obtain are two groups of nodes identifying the two different communities. This could be carried out by assigning different weights to the arcs and by computing two distinct eigenvalues. However, generally, it is difficult to assign "a priori" weights to links so that pages in the same community are ranked in some identifiable range.

Another problem arises because Kleinberg's algorithm finds the most relevant pages in the hypothesis of working on a connected *similarity graph*, associated to the *co-citation matrix* $A^{\mathrm{T}}A$, where $A$ is the adjacency matrix of the base set. However, since there could be more than one community, we must consider not only the principal eigenvector, but at least one other to obtain the two most relevant sets of authorities. Such an approach is expensive because it requires $k$ applications of the iterative procedure to obtain the ranking of $k$ different communities. So, a desirable property of a ranking algorithm should be the possibility of obtaining the correct result by executing the procedure just once, even in the case of unconnected graphs.


*Figure 1.* Example of a graph with $\lambda_1 = \lambda_2$.

### 2.3. The SALSA algorithm

The SALSA algorithm, proposed by Lempel and Moran [23], computes first the base set and then a random walk by alternatively (i) going randomly to a page which links to the current page, and (ii) going randomly to a page linked to by the current page. The authority (respectively hub) weights are defined by the stationary distribution of the two-step process, doing first step (i) and next step (ii) (respectively first step (ii) and next step (i)).

A nice property of the SALSA algorithm is that it is less affected than Kleinberg's algorithm of the *TCK* effect. A *Tightly-Knit Community* (*TCK*) is a small but highly interconnected set of pages. Consider two different communities, one $C_s$ with a small number of hubs and authorities, in which every hub points to all of the authorities and a much larger community $C_l$, in which the hubs point only to a subset of the authorities. In situations like this, the mutual reinforcement approach fails, giving a high rank to the pages of $C_s$ and a lower rank to the pages of $C_l$.

### 2.4. Other approaches and implementations

A different approach for computing authorities and hubs has been recently defined by Cohn and Chang [11]. They proposed a technique for the maximization of a *likehood* function based on a probabilistic model in which conditional distributions are considered. In particular, they postulate that there is a conditional distribution $P(c|z)$ of a citation $c$ given a topic $z$ and a conditional distribution $P(z|d)$ of a topic $z$ given a document $d$.

The use of random walks for ranking Web pages is also used in the PageRank algorithm. Indeed, the PageRank algorithm examines a single random walk on the global Web ranking the pages independently of the search query. With respect to Kleinberg's algorithm, the coupling between authorities and hubs is less tight and, consequently, the method is less vulnerable to the TCK effect.

The PageRank approach is the basis of the system *Google* which is currently the most successful search engine [17]. Several projects have implemented (variations of) Kleinberg's algorithm. We mention here the *HITS* technique [15] and *ARC* system [10]; the latter, besides the link structure analysis, also considers the text surrounding the hyperlinks in the pointing page (anchor text). The approaches considered in these projects, have been extended in the project *Clever* [20]. Further improvements of Kleinberg's algorithm have been proposed in [5].

A hybrid of the SALSA and PageRank algorithms, computing page reputations, has been presented in [27]. The algorithm works as follows: at each step, with probability $d$, the surfer jumps to a page of the collection chosen randomly, and with probability $1 - d$ he performs a SALSA step.

Moreover, the random walk approach was also adopted for measuring the quality of current search engines. In [19] Henzinger et al. proposed a measure for search engines, different from the traditional number of pages indexed. In particular, they provided an algorithm for approximating the quality of an index by performing a random walk on the Web, and used this methodology to compare the index quality of several major search

engines. An experiment-based criterion, for evaluating and comparing link analysis algorithms, is presented in [6].

## 3. Random walks

This section presents a new approach for topic distillation and community identification. Our idea is to obtain the different clusters directly from the co-citation matrix (associated to the *base* set obtained in the first step of Kleinberg's algorithm or of some algorithm based on the mutual reinforcement approach) without the application of the second step of Kleinberg's algorithm. This is carried out by denoting as the most relevant page the one which is the "most similar" to all the other pages in the same community. Our notion of similarity is based on the structure of the co-citation matrix: the "most similar" page, in a given community, has the property of having the greatest number of citations in common with the other pages in the community. So, the "most similar" page is also the most authoritative obtained by applying Kleinberg's algorithm. Thus, our technique identifies the most authoritative pages and it does not suffer from the previous mentioned problems.

To achieve this, we propose analyzing the behavior of a Random Walk over the graph associated with the co-citation matrix. Observe that such a graph is weighted and undirected and contains an arc between the node $i$ and $j$ with weight $w$ iff $i$ and $j$ are pointed to by the same set $S$ of pages and $|S| = w$. So the probability of going to a certain page $j$, starting from a page $i$, after a random walk through the other pages (following a random number of arcs) is proportional to the number of co-citations shared by all pages in such a path. The consequence is that this probability can be considered a measure of similarity. Before going into the details of our approach we provide some background on random walks.

*Definition 3.1.* Let $S$ be a set of states; a random walk on $S$ corresponds to a sequence of states. Moreover, it is *Markovian* if at each state the transition only depends on the current state (that is, it is not affected by the previous steps).

In our context, the states correspond to the different Web pages while the transitions are associated to the navigation of arcs (in the co-citation matrix). Since a transition between two states is only possible if there is an arc joining the associated pages in the similarity graph, a random walk defines a set of "similar" pages (connected in the similarity graph).

*Definition 3.2.* The equilibrium distribution of a state of a random walk is the fraction of the number of times the walk passes over the state and the total number of state transitions if it continues for an infinite time.

So, the first thing we need to define is the probability of a transition from one state to another, which corresponds to the probability of navigating a link of a Web page. This probability should take two factors into consideration: the information associated to the outgoing links (label, destination page, etc.) and the content of the source page (textual information); both link and node information are measured by means of a weights calculated

by using heuristics already present in the literature [10,17]. The information associated with the outgoing links says how interesting the linked pages are, whereas the textual information says how interesting the current page is.

The combination of this information gives the probability of navigating the different links and also the probability of remaining on the current page.

For modeling these two behaviors we define a transition probability matrix. Note that all the following definitions and theorems refer to a generic graph $G = (V, E)$; for our purpose this will be the graph computed in the first step of the mutual reinforcement algorithm.

*Definition 3.3.* Let $G = (V, E)$ be a weighted graph and $A$ the associated adjacency matrix where $A_{i,j}$ represent the weight of the arc $(i \to j) \in E$; let $C = A^{\mathrm{T}}A$ be the co-citation matrix and let $c_i$, with $i \in V$, be a weight associated to each node representing the textual information of the node. Assuming, by definition

$$C_{i,i} = c_i, \quad \forall i \in V,$$

the transition probability matrix $P$ is a $|V| \times |V|$ matrix in which every entry is

$$P_{i,j} = \frac{C_{i,j}}{\sum_{k=1}^{|V|} C_{i,k}}.$$

Observe that in the probability matrix $P$, $P_{i,i}$ denotes the probability of remaining in node $i$, whereas $P_{i,j}$, with $i \neq j$, denotes the probability of going from node $i$ to node $j$. We must emphasize the importance of defining the weights for arcs (weights $A_{i,j}$ with $i \neq j$) and pages (weights $A_{i,i}$); this task can be addressed by means of some heuristic already present in the literature. Anyhow, in our tests we do not adopt any heuristic and associate to arcs and pages the weight 1.

As seen, the matrix $P$ models the behavior of the unitary length transitions; its intuitive meaning is that $P_{i,j}$ represents the similarity between node $i$ and $j$ on the basis of one link only. Now to continue our analysis we want to extend this model in order to describe what happens in a random walk composed of more than one link.

*Definition 3.4.* Let $P$ be the transition probability matrix, and let $P^1_{i,j} = P_{i,j}$ the probability of going from node $i$ to node $j$ in one step, then the probability of going from node $i$ to node $j$ in $n$ steps is

$$P^n_{i,j} = P^{n-1}_{i,j} \times P_{j,j} + \sum_{k \neq i, k \neq j} P^{n-1}_{i,k} \times P_{k,j} + P^{n-1}_{i,i} \times P_{i,j}.$$

The intuitive meaning is that $P^n_{i,j}$ represents a measure of similarity between node $i$ and $j$ looking only at a co-citation through an $n$-dimensional path.

So, if we consider a random walk obtained for $n \to \infty$, what we will obtain is the equilibrium distribution of our random walk. In any case we observe that the walk length has

a probability distribution in itself. We decide to model this behavior with an exponential distribution. With this assumption the following definition is obtained.

*Definition 3.5.* Being in a node $i$, the probability of going to a node $j$ with a random walk of random length (composed with a maximum of $n$ steps) becomes

$$T_{i,j}(n) = (f-1) \times \left( \frac{1}{f} P^1_{i,j} + \frac{1}{f^2} P^2_{i,j} + \cdots + \frac{1}{f^n} P^n_{i,j} \right),$$

where $1/f$ is a damping factor, with $f > 0$.

Now we calculate the terms $T_{i,j}(n)$ for $n \to \infty$, so obtaining a value giving a measure of co-citations by considering paths (with an arbitrary number of links) instead of arcs (i.e., if the pages $i$ and $k$ point to a page $p$ and the pages $k$ and $j$ point to a page $q$, then $i$ and $j$ are indirectly coupled). Therefore, a higher value of $\sum_{i \in V} T_{i,j}(n)$ gives a measure of similarity of page $j$ with respect to all the other pages, that is, an high value of the sum means that page $j$ has many co-citations in common with all other pages; such a page is authoritative in the sense of Kleinberg theory. In the following, we shall denote with $T(n)$ the matrix in which every element with indexes $i$ and $j$ is $T_{i,j}(n)$. To apply this idea, we simply need to know the behavior of $T(n)$ for $n \to \infty$.

**Theorem 3.6.** Let $P$ be a matrix representing a transition probability and let $f$ be a real number greater than the principal eigenvalue of $P$, then the sequence

$$T(n) = (f-1) \times \sum_{i=1}^{n} \left( \frac{1}{f} \times P \right)^i$$

converges for $n \to \infty$ to the value

$$W = (f-1) \times P \times (f \times I - P)^{-1},$$

where $I$ is the identity matrix of the same size as $P$.

**Proof:** Observe that $T(n+1)$ can be rewritten in two way:

$$T(n+1) = T(n) + (f-1) \times \left( \frac{1}{f} \times P \right)^{(n+1)},$$

$$T(n+1) = T(0) + T(n) \times \left( \frac{1}{f} \times P \right),$$

where $T(0) = (f-1) \times (1/f \times P)$. Combining the two expression, we notice that $T(n)$ can also be calculated as

$$T(n) = (f-1) \times \left( \frac{P}{f} - \frac{P^{(n+1)}}{f^{(n+1)}} \right) \times \left( I - \frac{P}{f} \right)^{-1}.$$

To prove the convergence of the sum we can show that each element $P^{(n+1)}_{i,j}/f^{(n+1)}$ or, equivalently, each element $P^n_{i,j}/f^n$, for $n \to \infty$, converges to 0. In fact, as each element of $P^n$ is just a linear combination of terms like $\lambda^n$ at most multiplied by a term like $n^k$, by choosing $f > \lambda_1$ (i.e., by choosing $f$ to be greater than the principal eigenvalue), each element $P^n_{i,j}/f^n$ converges to 0. Then

$$T(n) = (f-1) \times \frac{P}{f} \times \left(I - \frac{P}{f}\right)^{-1} = (f-1) \times P \times (f \times I - P)^{-1}.$$

Finally, we can ensure that $(f \times I - P)$ is invertible since it has been assumed that $f$ is not an eigenvalue of $P$. □

### *3.1. Determining the value of f*

We have shown that our method converges if the term $f$ is made greater than the principal eigenvalue of the matrix $P$. There are two ways for achieving this. The first one requires a theoretical explanation and notation. Let $A$ be an adjacency matrix, then $G_A$ is the graph represented by $A$, called *support graph*.

*Definition 3.7.* The period of a graph $G$ is the greatest common divisor of the lengths of all cycles; if $G$ has period 1 we say that it is aperiodic. A matrix $P$ is aperiodic if the graph $G_P$ is aperiodic.

*Definition 3.8.* A matrix $P$ is irreducible if for every pair $i$, $j$ there is a path in the support graph of $P$ originating in $i$ and ending in $j$.

**Theorem 3.9.** If $P$ is irreducible and aperiodic the choice of $f = 2$ ensures the convergence of the formula in Theorem 3.6.

**Proof:** Recall that a real non-negative square matrix $P$ is *stochastic* if all its rows sum to 1 ($\sum_{j=1}^n P_{i,j} = 1$ for all $i$). Recall also that all eigenvalues of $P$ are $\leqslant 1$ and that 1 is an eigenvalue with eigenvector $[1, 1, \ldots, 1]^{\mathrm{T}}$ [4]. Consequently, being $P$ aperiodic and irreducible, we can apply the Ergodic Theorem [26] according to which the principal (simple) eigenvalue is $\lambda_1 = 1$. □

In practice we can calculate the period $p$ of the graph and then construct a virtual cycle (not associated with real pages) of length $p + 1$ to ensure the aperiodicity. Instead from the point of view of the irreducibility, we can create a node $v$ with no informative content pointing, with probability $1/n$, to all the other nodes. So, intuitively, this model simulates the behaviour of a surfer starting navigation from each node with the same probability. Moreover, adding links to $v$, originating in nodes without outlinks, corresponds to the situation in which such a surfer arriving in a node without leaving arcs decides to restart the navigation randomly. This construction ensures that $P$ is irreducible.

The second way for ensuring the convergence of our algorithm is based on the calculus of the principal eigenvalue $\lambda$ of $P$ with a simple procedure. Note that this task is simpler than the task of calculating the corresponding eigenvector. Then, we can simply choose $f = \lambda + 1$.

### 3.2. *Calculating the relevance*

Theorem 3.6 is the real kernel of our technique: the columns of the matrix $W$ can be used to obtain the relevance of pages. In fact, $W_j$ (column $j$ of $W$) contains the probabilities of arriving in the corresponding node $(j)$ with a random walk originating in nodes $i = 1, \ldots, |V|$, where $|V|$ is the size of the base set. So if we assume that the probability of being in node $i$ is $1/|V|$, the row vector

$$\overline{\pi} = \frac{1}{|V|} \times [1, \ldots, 1] \times W = \frac{f-1}{|V|} \times [1, \ldots, 1] \times P \times (f \times I - P)^{-1}$$

is such that each term $\overline{\pi}_i$ represents how much node $i$ is similar to the other nodes in the same community: that is the relevance of page $i$.

Observe that the vector $\Pi$ can be calculated, in an approximated way, by applying Definition 3.5, choosing a value of $n$ ensuring the desired precision (the convergence of the sum is, anyhow, guaranteed by Theorem 3.6). The formula, expressing the approximation of $\overline{\pi}$ (denoted by $\widetilde{\pi}$), is

$$\widetilde{\pi}(n) = \frac{f-1}{|V|} \times [1, \ldots, 1] \times \left( \frac{P^1}{f} + \frac{P^2}{f^2} + \cdots + \frac{P^n}{f^n} \right),$$

where $1/f$ is a damping factor, with $f$ chosen accordingly to Section 3.1. Denoting the vector $[1, \ldots, 1] \times P/f$ as $\overline{z}$, the formula can be rewritten as

$$\widetilde{\pi}(n) = \frac{f-1}{|V|} \times \overline{z} \times \left( I + \frac{P}{f} \times \left( I + \frac{P}{f} \times (\ldots) \right) \right),$$

where the multiplication operator is only applied to a row vector and a matrix (with $O|V|^2$ cost). The approximation $\widetilde{\pi}$ dependens on the value of $n$ and for $n \to \infty$ we get the exact solution $\overline{\pi}$. Again, the convergence of $\overline{\pi}$ is stated by Theorem 3.6.

Observe that our approach is not affected by the spectral structure of the matrix $W$, as the formula for calculating $\overline{\pi}$ still holds even if the graph is constituted by several strong components (i.e., it can be used to rank all authoritative pages belonging to different communities). Moreover, it is possible to weight the textual information associated with the links (depending on the query string) producing a better ranking of pages.

### 3.3. *Complexity*

Assume $G = (V, E)$ be the graph representing the base set, where $V$ is the set of pages and $E$ be the links in such a set. Our technique can be implemented by an algorithm with

time complexity $O(|V|^3)$ (the complexity of calculating the inverted matrix). Thus, our algorithm has the same complexity of other techniques based on the mutual reinforcement approach, for computing the principal eigenvalue (i.e., only one community). Moreover, to compute $t$ communities, the mutual reinforcement algorithm takes $O(t \times |V|^3)$, with $t = O(|V|)$. In practice, as Kleinberg uses an iterative procedure (with $k$ iterations), the first community is obtained in $O(k \times |V|^2)$ time and the first $t$ communities in $O(k \times t \times |V|^2)$ time. In any case, by fixing the value of $k$, we cannot be certain whether the chosen $k$ is big enough to ensure the validity of the result.

As observed in the previous subsection, our technique can be implemented by an iterative algorithm with the advantage of calculating all communities with $O(n \times |V|^2)$ cost and without inverting the matrix $(f \times I - P)$.

The computation of the approximate solution reduces the complexity and avoids the problems which can arise if the matrix is ill-conditioned.[4] However, considering the sparse (nearly bipartite) structure of the *base* set, the case of ill-conditioned matrices is not so frequent.

### *3.4. Discussion*

The novelty of our approach, with respect to other previous methods, is that it does not use the concept of random walk for describing the behavior of a user in surfing the Web. In particular, in the previous approaches, each page is considered as a state of a navigation, and the goal is to find those states in which it is more probable to lie after a random navigation of links. On the contrary, our approach does not make any assumption about user behavior, because it applies the notion of random walk to the co-citation graph and not to the Web graph. As a consequence, a link from page $i$ to page $j$ represents the number of citations that pages $i$ and $j$ have in common and not the probability of navigating the link $(i \rightarrow j)$. So, our approach identifies the relevant pages as those having the greatest number of common citations; this result, according to Kleinberg approach, gives the most authoritative pages.

## 4. System prototype

To verify the validity of our approach we have designed and developed a system prototype, called $\mathcal{STED}$ (a System for Topic Enumeration and Distillation), implementing the random walk approach presented in Section 3. The architecture of the prototype is reported in Figure 2.

As previously discussed there are two steps in our technique:

- First, it computes the *base* set, through a script written in the WebL language. This task is carried out by computing the *root* set, obtained by submitting the query string to a search engine (Alta Vista [13] in the current version) and, then, augmenting the root set into the base set. The result is stored in the database *DB*.

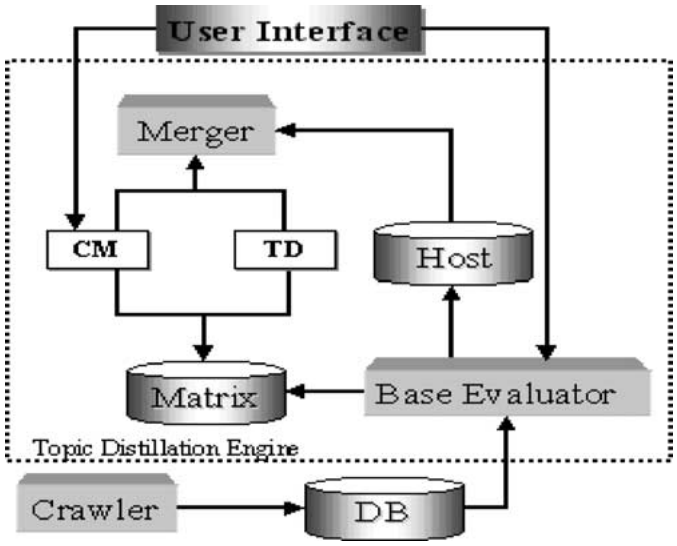


*Figure 2.* System architecture.

- Next, it applies our method to the *base* set, in order to obtain the vector $v$ described in Section 3.2. In this way it ranks the pages according to the $v_i$ values.

Moreover, in order to apply the above mentioned steps, it is necessary to first construct the database (called *DB* in Figure 2), associated to the whole Web. It stores information about both the textual content of Web pages and the link structure. In this respect, the system is similar to the Google search engine [7,8,17]. The database *DB* is constructed by a WebCrawler entirely written in the WebL language [12], a scripting language fully compatible with JAVA. The structures used to store information are of two different types: an inverted index storing textual information and a matrix (based on adjacency lists) storing links among pages.

The system can be seen by means of a user interface, which can be viewed in Figure 3 where the result for the query "abortion" is reported. The interface, interacting with the *Topic Distillation Engine*, which is the real kernel of the prototype, enables (i) the construction of the *base set* from the database *DB*, (ii) the computation of the authoritative pages, and (iii) the identification of the different communities with respect to the query topic.

In particular, the computation of the *base* set is performed by the *Base Evaluator* module which implements the first step of Kleinberg's algorithm [10,21], whereas the authoritative pages are computed by means of an algorithm implementing the Random Walk approach, through the modules *CM*, *TD* and *Merger*.

The *Base Evaluator* module interacts with the database *DB* and outputs the base set into two temporary files: *Host* and *Matrix*. The file *Host* contains information about pages, that is each page is associated with a unique numeric *id* and with a weight representing the page's informative content with respect to the supplied query. The file *Matrix* contains


*Figure 3.* User interface.

the adjacency matrix $A$, where each node (i.e., Web page) is denoted by the identifier associated with the page in the *Host* file.

With respect to other approaches, this one combines the link analysis with a content page evaluation in order to consider the probability of remaining on a certain page if it is relevant enough for the user.

More specifically, each tuple of the file *Matrix*, representing an arc in the Web graph, is in the form $\langle source, destination, weight \rangle$, in which *weight* is obtained evaluating the relevance of the links for the query (i.e., how the query string terms appear in the anchor text, how links are relevant according to their positions, etc.). Therefore, the matrix $A$ is submitted to two modules working in parallel: the first one, called *CM* (*Clustering Module*), calculates the components of the co-citation matrix associated to $A$, while the second one, called *TD* (*Topic Distillation* module), performs the searching of relevant pages. Finally, the *Merger* module takes the outputs of the modules *CM* and *TD*, the *Host* file and groups the pages, ranked by relevance, according to the communities computed by the *CM* module.

The real kernel of this technique lies in the Topic Distillation block that performs the ranking calculus on the basis of the results presented in Section 3.2.

## 5. Experimental results

We tested the prototype with base sets of different sizes, and compared our results with those supplied by Kleinberg's algorithm. Using a *base* set of about 2000 pages, for sets defining a unique community, we found no relevant differences in the results. For instance, consider the pages obtained in reply to the query string "search engines," summarized in Table 1.

*Table 1.* Comparison with Kleinberg's approach for query "search engines"

| Ranking | Kleinberg | $\mathcal{STED}$ |
|---|---|---|
| 1 | www.highway61.com | www.yahoo.com |
| 2 | www.yahoo.com | www.highway61.com |
| 3 | argos.evansville.edu | www.excite.com |
| 4 | www.excite.com | www.mckinley.com |
| 5 | www.mckinley.com | www.malaysiadirectory.com |
| 6 | www.lycos.com | www.change.org |

*Table 2.* Comparison with Kleinberg's approach for query "JAVA"

| Ranking | Kleinberg | $\mathcal{STED}$ |
|---|---|---|
| 1 | www.javaarchives.com | java.sun.com |
| 2 | ads.aceweb.net | www.javaworld.com |
| 3 | www.ScreenSaverArchives.com | www.java-pro.com |
| 4 | www.NTWare.com | www.microsoft.com |

The comparison of the results shows something interesting when there is more than one community and a base set smaller than usual is chosen. In Table 2 we reported the results for the query "JAVA" with a *base* set of just 500 nodes.

We point out that in this case our statistical approach performs better than Kleinberg's technique, even if applied to a small collection of nodes; that is, it seems to be less affected by the dimension of the input.

Another interesting aspect of our approach is that it can be used even if there are many communities; in this case Kleinberg supplies only the community associated to the principal eigenvector, while our method ranks all nodes; so we obtain pages belonging to different communities ranked together. As an example, let us consider the results to the query "abortion", shown in Table 3. In the Web, with regard to this query, there are three main types of communities: pro-life sites, pro-choice sites and sites encouraging abortion. As the size of the pro-life community is the greatest, using Kleinberg's algorithm we obtain pages belonging to this community only.

However, as can be seen in Table 3, $\mathcal{STED}$ supplies pages belonging to all the three identified communities. In fact, `www.prochoice.com`, `www.naral.com` and `www.cais.com` are pro-choice pages, `www.gynpages.com` and `www.abortion-help.com` are pages in which one can find useful information on abortion, while all the others are pro-life pages.

As a further example, in reply to the query "jaguar" we obtain several communities, the most relevant of which are reported in Table 4. These are respectively associated to "jaguar" as the car manufacturer and as the football team.

Our prototype has also been tested with the query "Java OR Search Engine" whose result contains (obviously) pages belonging to different communities and, as usual, Kleinberg's algorithm supplies only the largest one. Using $\mathcal{STED}$, instead, sites belonging to both groups are returned; moreover, we can distinguish the two communities by just analyzing the *base* set.

*Table 3.* Comparison with Kleinberg's approach for query "abortion"

| Ranking | Kleinberg | $\mathcal{STED}$ | Comm. |
|---|---|---|---|
| 1 | www.nrlc.org | www.nrlc.org | 1 |
| 2 | www.prolife.org | www.prolife.org | 1 |
| 3 | members.aol.com/pladvocate/ | www.ohiolife.org | 1 |
| 4 | www.rtl.org | www.naral.org | 2 |
| 5 | www.peopleforlife.org | www.prochoice.org | 2 |
| 6 | www.ohiolife.org | www.peopleforlife.org | 1 |
| 7 | www.hli.org | www.hli.org | 1 |
| 8 | www.heritagehouse76.com | www.rtl.org | 1 |
| 9 | www.lifecall.org | www.thenewamericans.org | 1 |
| 10 | www.serve.com | www.gynpages.com | 3 |
| 11 | www.pregnancycenters.org | www.pregnancycenters.com | 1 |
| 12 | members.tripod.com | www.heritagehouse76.org | 1 |
| 13 | www.pfli.org | www.abortion-help.com | 3 |
| 14 | www.afterabortion.org | www.religioustolerance.org | 1 |
| 15 | www.abortionalternatives.com | www.cais.com | 2 |

*Table 4.* Communities 1 and 2 for query "jaguar"

| Ranking | Site |
|---|---|
| 1 | autos.yahoo.com |
| 2 | www.jaguardealer.com |
| 3 | www.jec.org.uk |
| 4 | www.gtjaguar.com |
| 5 | www.xks.com |
| 6 | www.us.jaguar.com |
| 7 | www.jaguarcars.com |
| 8 | www.scottsdalejag.com |
| 9 | www.manhattanjagrkvl.com |
| 10 | www.jcna.com |
| 1 | www.footballfanatics.com |
| 2 | jaguars.jacksonville.com |
| 3 | www.nfl.com |
| 4 | www.netsportmag.com |
| 5 | www.macjag.com |

The search of documents containing the string "Università italiane" is another interesting query, for which we expect to obtain the most important Italian Universities; obviously, in this case, the system returns one community only (reported in Table 5).

The really interesting aspect of our approach is that it ranks sites belonging to different communities by applying the algorithm only once; on the contrary, Kleinberg's approach needs to calculate the other non-principal eigenvectors (i.e., the algorithm must be applied several times). In particular, once all the relevant pages are obtained, in order to identify the community to which each page belongs, $\mathcal{STED}$ groups the results by simply performing a visit of the graph associated with the co-citation matrix. Moreover, in our approach

*Table 5.* Result for the query "Università italiane"

| Ranking | Site |
|---|---|
| 1 | www.unina.it/ |
| 2 | www.unipd.it/ |
| 3 | www.unibo.it/ |
| 4 | www.unimi.it/ |
| 5 | www.unipv.it/ |
| 6 | www.univ.trieste.it/ |
| 7 | www.unipi.it/ |
| 8 | www.polimi.it/ |
| 9 | www.uniroma1.it/ |

no heuristic is needed for ranking pages of different communities; in fact the smaller the community, the smaller each term $v_i$ will be, for each node $i$ of such a community (see Section 3.2). This happens because most of the pages (outside the small community) do not contribute to the transition probability, as there is no path for going inside the small community.

## 6. Conclusion

In this paper, we have presented a new approach for Topic Distillation on the Web which computes authoritative pages by analyzing the structure of the base set. The technique applies a statistical approach to the co-citation matrix (of the base set) to find the most co-cited pages and combines the link analysis with a content page evaluation to consider the probability of remaining on a certain page if it is relevant enough for the user. We have shown that our technique is more efficient than other techniques, based on the mutual reinforcement method, previously proposed in the literature and provided several experiments showing its validity.

The proposed approach has several interesting properties:

1. it can be applied to unconnected graphs without additional costs,
2. in the approximated form, it does not need to make matrix inversion and has no convergence problem,
3. in the case of a not ill-conditioned matrix, it can be implemented by using a closed form.

## Notes

1. Generally, $\Sigma$ can be the alphabet of any closed semiring.
2. Based on the mutual reinforcement relationship holding among hubs and authorities.
3. Assuming the same weight for all arcs, the adjacency matrix has one eigenvalue with multiplicity two.
4. The inversion of a matrix is a problematic task in the case it is ill-conditioned.

## References


[1] S. Abiteboul, D. Quass, J. McHugh, and J. Widom, "The Lorel query language for semistructured data," *Internat. J. Digital Libraries* 1(1), 1997, 68–88.

[2] V. Apparao et al., "Document object model (DOM) level 1 specification version 1.0," 1998, http://www.w3.org/TR/REC-DOM-level-1.

[3] R. Baeze-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, Reading, MA, 1999.

[4] R. Bellman, *Introduction to Matrix Analysis*, SIAM, Philadelphia, PA, 1997.

[5] K. Bharat and M. R. Henzinger, "Improved algorithms for topic distillation in a hyperlinked environment," in *Proc. of ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1998.

[6] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, "Finding authorities and hubs from link structures on the World Wide Web," in *Proc. of WWW Conference*, 2001, pp. 415–429.

[7] S. Brin and L. Page, "The PageRank citation ranking: Briging order to the Web, http://google.standford.edu/backrub/pageranksub.ps.

[8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," in *Proc. of the 7th Internat. WWW Conference*, 1997.

[9] S. J. Carrire and R. Kazman, "WebQuery: Searching and visualizing the Web through connectivity," *Computer Networks* 29(8–13), Special Issue on 6th Internat. WWW Conference, 1997, 1257–1267.

[10] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan, "Automatic resource list compilation by analizing hyperlink structure and associated text," in *Proc. of the 7th Internat. WWW Conference*, 1998, pp. 65–74.

[11] D. Cohn and H. Chang, "Learning to probabilistic identify authoritative documents," Technical Report, 2000.

[12] Compaq Computer Cooperation, http://www.research.digital.com/webl/.

[13] Digital Equipment Corporation, "AltaVista Search Engine," http://www.altavista.com/.

[14] C. Dwork, S. R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the Web," in *Proc. of the 10th Internat. WWW Conference*, 2001, pp. 613–622.

[15] D. Gibson, J. M. Kleinberg, and P. Raghavan, "Inferring Web communities from link topology," in *Proc. of the 9th ACM Conf. on Hypertext and Hypermedia*, 1998.

[16] R. Goldman, J. McHugh, and J. Widom, "From semistructured data to XML: Migrating the Lore data model and query language," *Internat. Workshop on the Web Databases*, 1999, pp. 25–30.

[17] Google Corporation, Google search engine, http://www.google.com.

[18] G. Greco, S. Greco, and E. Zumpano, "A probabilistic approach for discovering authoritative Web pages," in *Proc. of the 2nd Internat. Conf. on Web Information Systems Engineering*, Kyoto, Japan, 2001.

[19] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, "On near-uniform URL sampling," *Computer Networks* 33(1–6), Special Issue on 9th WWW Conference, 2000, 295–308.

[20] IBM Corporation Almaden Research Center, *Clever*, http://www.almaden.ibm.com/cs/k53/clever.html.

[21] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proc. of the 9th ACM–SIAM Symposium on Discrete Algorithms*, 1998.

[22] R. Kumar, P. Taghavan, S. Rajagopalan, and D. Sivakumar, "The Web as a graph," in *Proc. of the 19th ACM Symposium on Principles of Database Systems*, 2000, pp. 1–10.

[23] R. Lempel and S. Moran, "The stochastic approach for link-structure analysis (SALSA) and the TCK effect," in *Proc. of the 7th Internat. WWW Conference*, 1998.

[24] M. Marchiori, "The quest for correct information on the Web: Hyper search engines," *Computer Networks* 29(8–13), Special Issue on 6th Internat. WWW Conference, 1997, 1225–1236.

[25] P. Pirolli, J. E. Pitkow, and R. Rao, "Silk from a Sow's ear: Extracting usable structure from the Web, in *Proc. of the 9th ACM–SIGCHI Conference*, 1996, pp. 118–125.

[26] M. Pollicott, and M. Yuri, *Dynamical Systems and Ergodic Theory*, Cambridge Univ. Press, Cambridge, 1998; on line version at http://www.maths.man.ac.uk/mp/book.html.

[27] D. Rafiei and A.O. Mendelzon, "What is this page known for? Computing Web page reputation," *IEEE Data Engineering Bulletin* 23(3), 2000, 9–16.

[28] H. Small, "Co-citation in scientific literature: A new measure of the relationship between two documents," *J. American Soc. Info Sci.*, 1973, 275–279.
[29] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton Univ. Press, Princeton, 1994.