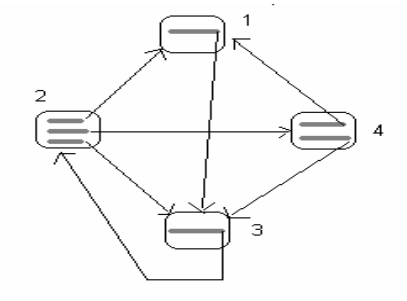
WEB RETRIEVAL AND RANKING

11.1 Web Graph

Let $W_1,\ldots,W_i,\ldots,W_N$ denote a set of Web pages. A directed link from page W_i to page W_j is defined by the fact that the URL of page W_j occurs on page W_i; notation: $W_i \rightarrow W_j$.

A graph $G = (V, E)$ is referred to as a *Web graph* if vertex $v_i \in V$ corresponds to page W_i ($i = 1,\ldots,N$), and a directed edge $(v_i, v_j) \in E$ exists if there is a link from page W_i to page W_j. Graph G can be represented, e.g., by an adjacency matrix $M = (m_{ij})_{N\times N}$ defined as follows (Fig 11.1):

$$m_{ij} = \begin{cases} 1 & W_i \rightarrow W_j \\ 0 & otherwise \end{cases} \qquad (11.1)$$



M

0	0	1	0
1	0	1	1
0	1	0	0
1	0	1	0

Fig. 11.1. A small Web graph G with four pages: 1, 2, 3 and 4. The horizontal bars within each page symbolise URLs indicating links to other pages as shown by the arrows. The corresponding adjacency matrix M is shown on the right

The number of outgoing links from page W_i is called the *outdegree* of page W_i. The number of incoming links to page W_i is called the *indegree* of page W_i. For example, in Fig 11.1, the outdegree of page 2 is equal to 3, while its indegree is equal to 1.

11.2 Link Structure Analysis

Link structure analysis (*link analysis*, for short) refers to methods used to quantify importance of networked entities under focus based on the number of links (connections) between them. Entities may be

- social objects (e.g., groups of people),
- written units (e.g., scientific papers),
- Web pages,
- molecules,
- etc..

The starting point of link analysis was *citation analysis* whose principle is as follows: the number of citations a paper gets from other papers is a measure of its importance (Garfield 1955, 1972).

This principle was applied to compute an *impact factor* for journals. For example, the impact factor *IF* for journal *J* in 2007 can be calculated as follows:

Impact Factor Method

$$IF = \frac{C}{P},$$

where *C* is the number of citations *J*'s articles published in 2005 and 2006 get from other journals during 2007, and *P* is the number of articles published in *J* during 2005 and 2006.

The impact factor is based merely on a pure count of links; no other factor (e.g., quality, importance) is being taken into account.

The principle of citation analysis was applied for the first time in (Carriere and Kazman 1997) for the Web retrieval, in the form of the following method:

Connectivity Method

1. Using the Boolean retrieval method, a list of Web pages is obtained first (hit list).
2. The Web graph for the hit list is constructed.
3. For each node in the graph, its *connectivity* (i.e., the sum of its indegree and outdegree) is computed.
4. Finally, the hit list is sorted on node connectivity, and presented in decreasing order.

Pinski and Narin enhanced the Connectivity Method by noting that not all citations have equal importance. They argued that a journal is important if it gets citations from other important journals (Geller 1978). The Mutual Citation Method proposed is as follows:

Mutual Citation Method

Let $J_1,\ldots,J_i,\ldots,J_n$ denote entities. A matrix $M = (m_{ij})_{n\times n}$ is constructed as follows:

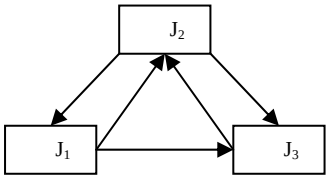
$$m_{ij} = \frac{c_i^j}{c_i},$$

where c_i denotes the total number of citations in journal J_i, while c_i^j denotes the number of citations journal J_j gets (out of c_i) from journal J_i. The importance vector $\mathbf{w} = (w_1 \ \ldots \ w_n)$ of journals is the solution of the following equation:

$$\mathbf{w} = M^{\mathrm{T}}\mathbf{w}.$$

In other words, the importance vector **w** is the eigenvector corresponding to eigenvalue 1 of matrix M^T.

Example 11.1.
Let us consider the following small Web graph:



Matrix M is as follows:

$$\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

The importance w_1 of page J_1 is equal to

$$w_1 = 0 \cdot w_1 + 0.5 \cdot w_2 + 0.5 \cdot w_3,$$

where (0 0.5 0.5) is the first row of matrix M. So, the importance vector $\mathbf{w} = [w_1\ w_2\ w_3]^T$ is given by the equation $\mathbf{w} = M^T\mathbf{w}$ and is equal to $\mathbf{w} = [0.371; 0.743; 0.557]^T$. □

11.3 The PageRank Method

In the PageRank method, a Web page's importance is determined by the importance of Web pages linking to it. Brin and Page (Brin and Page 1998) define the PageRank value R_i of a Web page W_i using the following equation:

$$R_i = \sum_{W_j \in B_i} \frac{R_j}{L_j}, \qquad (11.10)$$

where L_j denotes the number of outgoing links (i.e., URLs) from page W_j, and B_i denotes the set of pages W_j pointing to page W_i.

Eq. (11.10) is a homogenous and simultaneous system of linear equations in the unknown R_i, $i = 1,\ldots,N$, which always has trivial solution (the null vector, i.e., $R_i = 0$, $i = 1,\ldots,N$).

Eq. (11.10) has nontrivial solutions too if and only if its determinant is equal to zero. Let $G = (V, A)$ denote (correspond to) a Web graph, where the set $V = \{W_1,\ldots,W_j,\ldots,W_N\}$ of vertices denotes the set of Web pages. The set A of arcs consists of the directed links (given by URLs) between pages.

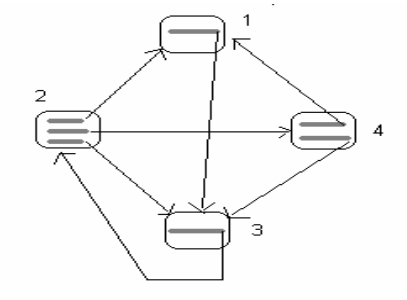
Let $M = (m_{ij})_{N\times N}$ denote a square matrix (modified adjacency matrix) attached to graph G such that (Fig. 11.4):

$$m_{ij} = \begin{cases} \dfrac{1}{L_j} & W_j \rightarrow W_i \\ 0 & \textit{otherwise} \end{cases}. \qquad (11.11)$$

Because the elements of matrix M are the coefficients of the right hand side of eq. (11.10), this can be re-written in a matrix form as follows:

$$M \times R = R, \qquad (11.12)$$

where R denotes the vector (i.e., column matrix) of PageRank values, i.e.,



M

0	1/3	0	1/2
0	0	1	0
1	1/3	0	1/2
0	1/3	0	0

Fig. 11.4. A small Web graph G with four pages: 1, 2, 3 and 4. The elements of matrix M are also shown, they were computed using (11.11)

$$R = \begin{bmatrix} R_1 \\ . \\ R_i \\ . \\ R_N \end{bmatrix} = [R_1, \ldots, R_i, \ldots, R_N]^{\mathsf{T}}. \qquad (11.13)$$

If graph G is strongly connected (i.e., every node can be reached from any other node following directed links), the sums of the columns in matrix M are equal to 1. Thus, because matrix M has only zeroes in the main diagonal, in matrix $M - I$ (I denotes the unity matrix), i.e.,

$$M - I = \begin{bmatrix} -1 & \ldots & m_{1N} \\ . & . & . \\ m_{N1} & \ldots & -1 \end{bmatrix}, \qquad (11.14)$$

the sums of columns is equal to zero. Let D denote its determinant, i.e.,

$$D = \left| M - I \right|. \qquad (11.15)$$

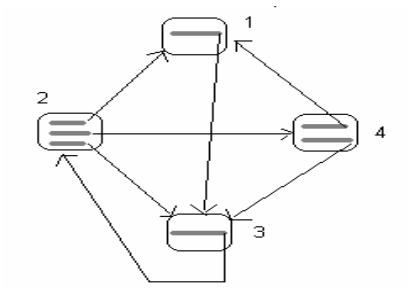
If every element of, for example, the first line of D is doubled, we obtain a new determinant D', and we have $D' = 2 \times D$. We add now, in D', every line to the first line. Because the sums of the columns in D

are null, it follows that (after these additions) the first row of determinant *D'* will be equal to the first row of determinant *D*. Thus, we have:

$$D' = 2D = D, \qquad (11.16)$$

from which it follows that $D = 0$. Because matrix $M - I$ is exactly the matrix of eq. (11.10), it follows that it has nontrivial solutions too.

The determinant $|M - I|$ being equal to 0 means that the number 1 is an eigenvalue of matrix *M*. Moreover, the number 1 is a *dominant eigenvalue* of matrix *M*, i.e., it is the largest eigenvalue, in terms of absolute value (Farahat et al 2006). Fig. 11.5 shows an example for the Web graph of Fig. 11.4.



M					*R*
0	1/3	0	1/2		0.325
0	0	1	0		0.651
1	1/3	0	1/2		0.651
0	1/3	0	0		0.217

Fig. 11.5. A small Web graph *G* with four pages: 1, 2, 3 and 4. The elements of matrix *M* are also shown, they were computed as follows: $m_{ij}=1/L_j$. The PageRank values, i.e., the eigenvector corresponding to eigenvalue 1, were computed using the Mathcad command 'eigenvec(*M*,1)'

The PageRank values are computed in practice (due to the fact that N is large) using some numeric approximation procedure by calculating the eigenvector R corresponding to eigenvalue 1. The following approximation method can be used:

$$M \times R_k = R_{k+1}, \;\; k = 0, 1, \ldots, K,$$
$$R_0 = \left[\frac{1}{N} \quad \ldots \quad \frac{1}{N} \right], \tag{11.17}$$

where K is equal to a few tens (typically to 50), or the recursive computation is performed until

$$\max | R_{k+1} - R_k | < \varepsilon, \tag{11.18}$$

where $\varepsilon \in \mathbb{R}_+$ is some preset error threshold. The approximation obtained is an eigenvector whose elements sum to unity.

Example 11.2.

For the Web graph of the Fig. 11.5, the numerical approximation procedure yields the following PageRank values for $\varepsilon = 0.1$ and after $k = 9$ steps: $[0.177; 0.353; 0.35; 0.12;]^T$. □

Eqs. (11.17) are derived from the well-known *power method* used to compute the dominant eigenvector **x** of a matrix *M*, in general. The steps of the power method are as follows:

POWER METHOD

1. Choose an initial vector $\mathbf{x}_0$.
2. Set $i = 1$.
3. Calculate the next approximation $\mathbf{x}_{i+1}$ as $\mathbf{x}_{i+1} = M\mathbf{x}_i$.
4. Divide $\mathbf{x}_{i+1}$ by its Euclidean norm, i.e., $\mathbf{x}'_{i+1} = \frac{\mathbf{x}_{i+1}}{\|\mathbf{x}_{i+1}\|}$. *Note*. One may divide by any non-zero element of $\mathbf{x}_{i+1}$.
5. Repeat steps 3 and 4 until $error(\mathbf{x}'_i, \mathbf{x}'_{i+1}) < \varepsilon$, where $error(\mathbf{x}'_i, \mathbf{x}'_{i+1}) = ||\mathbf{x}'_i - \mathbf{x}'_{i+1}||$, or $error(\mathbf{x}'_i, \mathbf{x}'_{i+1}) = \max|\mathbf{x}'_i - \mathbf{x}'_{i+1}|$, or some other expression (as more appropriate for the application under focus).
6. The dominant eigenvector can be approximated by the Rayleigh-quotient: $\frac{\mathbf{x}^T M\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$.

For a real portion of the Web, graph *G* is not always strongly connected. For example, it may happen that a page W_j does not have any outgoing links (i.e., its outdegree is null). Such a page is referred to as a *dangling page*. In such a case, the *j*th column — corresponding to page W_j — of matrix *M* contains only zeroes. The elements of matrix *M* may be interpreted in the following way: the entry m_{ij} is the probability with which page W_i follows page W_j during a walk on the Web (i.e., the probability with which, during a navigation on the Web, a surfer jumps from page W_j to page W_i). Based on this interpretation, a new matrix, denoted by *M'*, can be constructed as follows:

1. Firstly, the columns corresponding to dangling nodes in matrix M are replaced by columns containing all $1/N$, i.e.,

$$m'_{ij} = \frac{1}{N}, \quad i = 1,\ldots,N, \text{ page } W_j \text{ is a dangling page.} \tag{11.19}$$

2. Secondly, using matrix M', a new matrix, M'', is computed as follows:

$$M'' = \alpha M' + (1 - \alpha)M', \quad 0 < \alpha < 1. \tag{11.20}$$

A typical value for α is $\alpha = 0.85$. Thus, the PageRank equation becomes:

$$M'' \times R = R. \tag{11.21}$$

Matrix M'' is nonnegative (i.e., its elements are nonnegative numbers), hence it has a nonnegative dominant eigenvalue (Farahat et al 2006). The corresponding eigenvector is the PageRank vector: it is unique, its entries are nonnegative, and it can be calculated using the approximation (or power) method given by eq. (11.17).

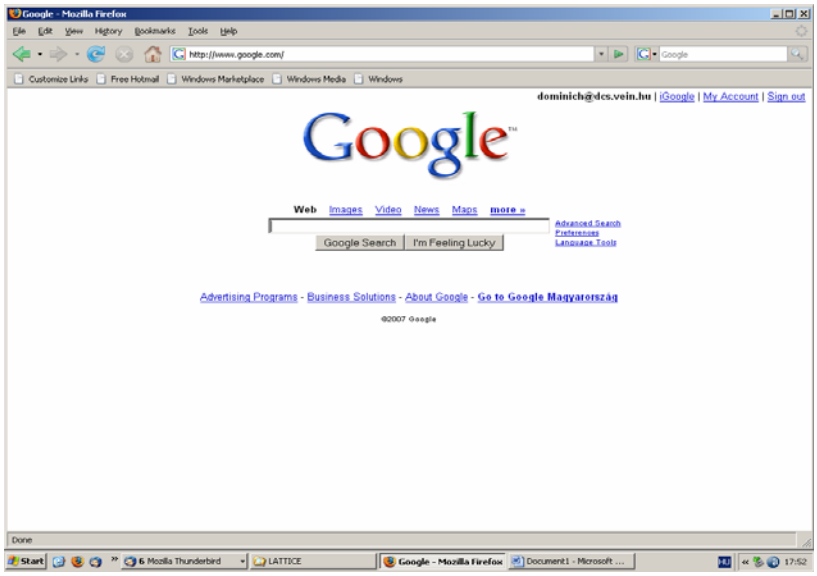
Example 11.3.
Let us assume that, in Fig. 11.5, page W_3 is a dangling page. Then,

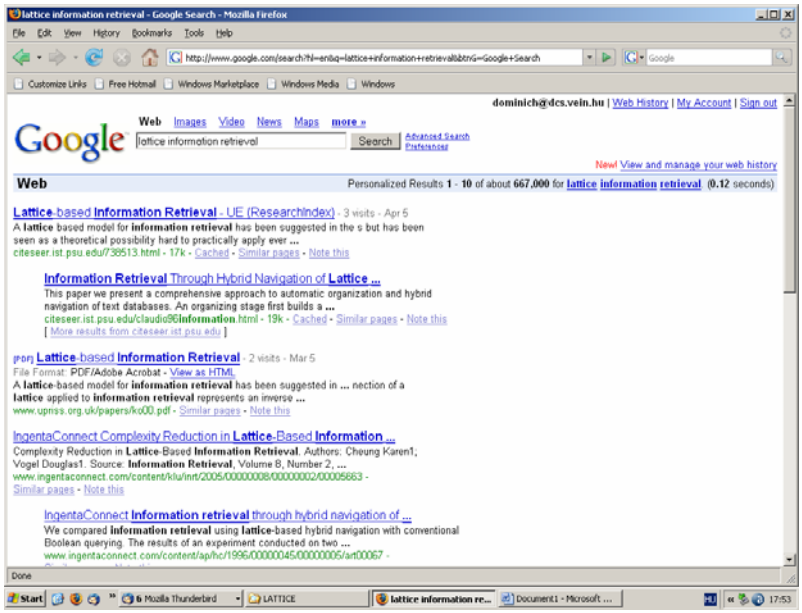
$$M'' = 0.8M' + 0.2M' = \begin{pmatrix} 0 & 0.333 & 0.25 & 0.5 \\ 0 & 0 & 0.25 & 0 \\ 1 & 0.333 & 0.25 & 0.5 \\ 0 & 0.333 & 0.25 & 0 \end{pmatrix},$$

whose dominant eigenvalue is 1, while the corresponding eigenvector, i.e., the PageRank vector, is $[0.419 \quad 0.21 \quad 0.838 \quad 0.279]^T$. □

11.3.1 Application of the PageRank Method in Web Retrieval

The PageRank method is being used by the Web search engine Google. Fig. 11.6 shows the query interface and a portion of the hit list for the query “lattice information retrieval” (as of the 2nd of May 2007).



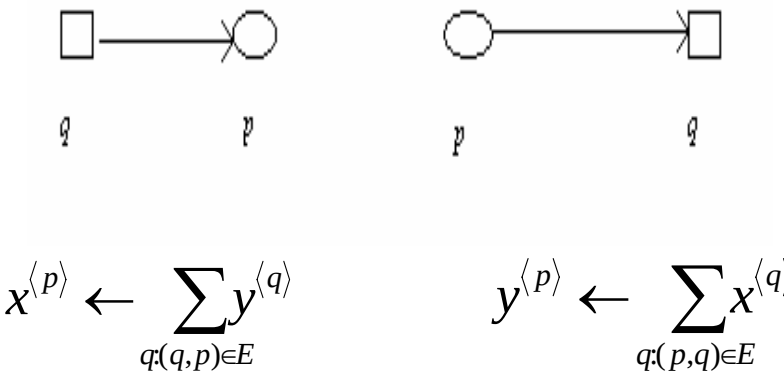


11.4 The HITS Method

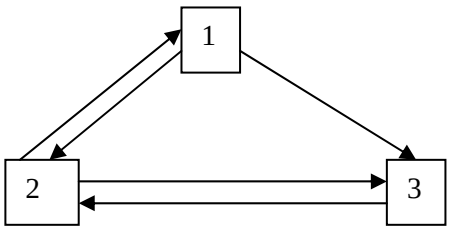
A method, called HITS, for computing hubs and authorities is proposed in (Kleinberg, 1999). Two types of Web pages are defined first: hubs and authorities. They obey a mutually reinforcing relationship, i.e., a Web page is referred to as

- an *authority* if it is pointed to by many *hub* pages,
- and a *hub* if it points to many *authoritative* pages (Fig. 11.7.a).

Given a page p, an authority weight $x^{<p>}$ and a hub weight $y^{<p>}$ is assigned to it. If p points to pages with large x-values, then it receives large y-values, and if p is pointed to by pages with large y-values, then it should receive a large x-value.



a)



b)

Fig. 11.7. a) Illustration of operations for computing hubs and authorities. **b)** Mini-Web (example)

The following iterative operations are defined:

$$x^{\langle p \rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q \rangle}$$
$$y^{\langle p \rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q \rangle}, \qquad (11.22)$$

where E denotes the set of arcs of the Web graph. Let M denote the adjacency matrix of the Web graph under focus. Then, eqs. (11.22) can be written in matrix form as follows:

$$x^{\langle k \rangle} = M^{\mathsf{T}} M\, x^{\langle k-1 \rangle},$$
$$y^{\langle k \rangle} = M M^{\mathsf{T}} y^{\langle k-1 \rangle}. \qquad (11.23)$$

Matrix $M^{\mathsf{T}}M$ is referred to as *hub matrix*, while matrix MM^{T} as *authority matrix*. Thus, the HITS method is equivalent to solving the following eigenvector problems:

$$M^{\mathsf{T}}Mx = \lambda x,$$
$$MM^{\mathsf{T}}y = \lambda y, \qquad (11.24)$$

where λ denotes the dominant eigenvalue of $M^{\mathsf{T}}M$ (MM^{T}). An entry m_{ij} in matrix MM^{T} is equal to the number of pages to which both pages i and j point. An entry m_{ij} in matrix $M^{\mathsf{T}}M$ is equal to the number of pages which point to both pages i and j. A diagonal entry (i, i) in matrix MM^{T} represents the outdegree of page i. In order to compute the authority and hub vectors in practice, the following steps are performed:

HITS Method

1. Define a root set S of Web pages (e.g., as follows: submit a query on some topic to a commercial search engine, and keep the top L hits).

2. Expand the root set S with the pages given by the inlinks and outlinks of pages in S to obtain a base set T.

3. Eliminate pages having the same domain name.

4. Define the Web graph for the base set T.

5. A sufficient number of iterations are repeated starting with the initial values $x_0 = [1,\ldots,1]^T$ and $y_0 = [1,\ldots,1]^T$ for both x and y, as follows:

   $$x_{i+1} = M^T y_i, \quad y_{i+1} = M x_{i+1};$$

   (after each iteration vectors x and y are normalised such that the squares of their entries sum to 1; this operation is called length normalisation).

It can be shown that x is the dominant eigenvector of M^TM, and y is the dominant eigenvector of MM^T (Farahat *et al* 2006).

Example 11.4.

Let

$$M = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

denote the adjacency matrix of the mini-Web graph of Fig. 11.7.b). We have:

$$MM^{\mathsf{T}} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

and

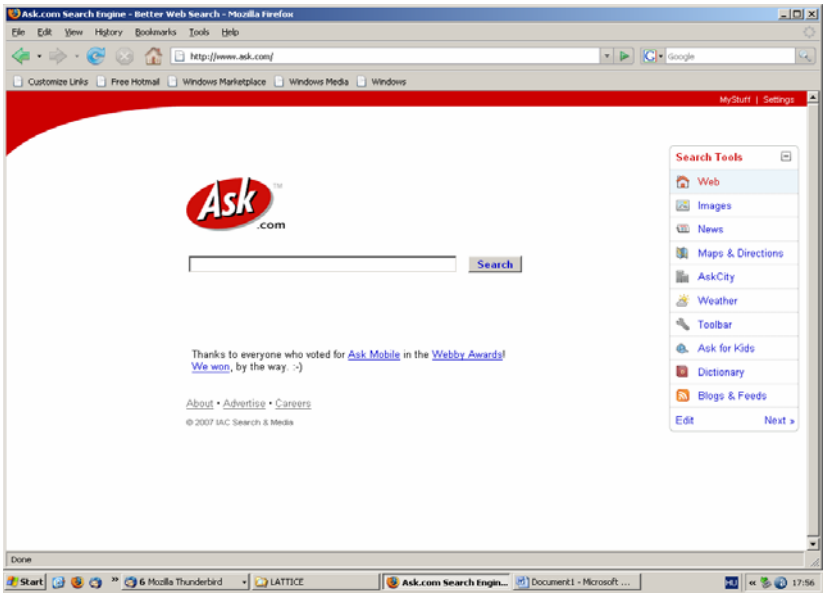
$$M^{\mathsf{T}}M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

with the following eigenvalues: 1.5, 0.2, 3.2. Perform the operations $x_{i+1} = M^{\mathsf{T}}y_i$ and $y_{i+1} = Mx_{i+1}$ until vectors x and y do not change significantly (convergence). In this example, after three steps, the following values are obtained: $x = [0.309; 0.619; 0.722]^{\mathsf{T}}$ and $y = [0.744; 0.573; 0.344]^{\mathsf{T}}$. □

11.4.1 Application of the HITS Method in Web Retrieval

The HITS method is being applied in the Web search engine Teoma (Ask)[1]. Fig. 11.8 shows the interface screen and a hit list screen for the query “lattice information retrieval” (as of the 2nd of May 2007).

[1] http://www.ask.com



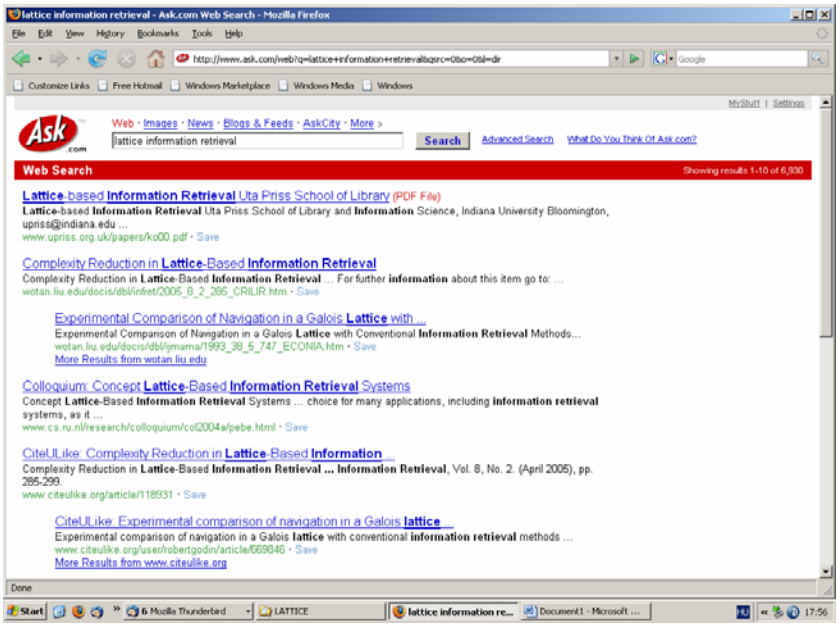


Fig. 11.8. Interface screen and a hit list of the Web search engine Ask (Teoma) which applies the HITS method

11.5 The SALSA Method

The SALSA method (Lempel and Moran 2001) offers another computation of authorities and hubs. Let $M = (w_{ij})_{N\times N}$ denote the adjacency matrix of the Web graph under focus. Let $M_r = (r_{ij})$ and $M_c = (c_{ij})$ be the following matrices (Langville and Meyer 2005):

$$\begin{aligned} r_{ij} &= \frac{w_{ij}}{\sum_{j=1}^{n} w_{ij}}; \quad \sum_{j=1}^{n} w_{ij} \neq 0; \quad i, j = 1,..., N \\ c_{ij} &= \frac{w_{ij}}{\sum_{i=1}^{n} w_{ij}}; \quad \sum_{i=1}^{n} w_{ij} \neq 0; \quad i, j = 1,..., N \end{aligned} \tag{11.26}$$

Two matrices, H (*hub matrix*) and A (*authority matrix*), are introduced as follows:

$$\begin{aligned} H &= M_r \times M_c^{\mathsf{T}}, \\ A &= M_c^{\mathsf{T}} \times M_r. \end{aligned} \tag{11.27}$$

The hub weights and authority weights are the elements of the dominant eigenvectors of H and A, respectively.

Example 11.6.
Using Fig. 11.7, we have:

$$M = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad M_r = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}, \quad M_c = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{pmatrix}.$$

Then,

$$H = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.75 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix}, \text{ and } A = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0 & 0.75 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}.$$

The dominant eigenvalue of H is 1, while the hub vector is the corresponding eigenvector: $[0.577\ 0.577\ 0.577]^{\mathsf{T}}$. The dominant eigenvalue of A is 1, while the authority vector is the corresponding eigenvector: $[0.577\ 0.577\ 0.577]^{\mathsf{T}}$. □

Originally, the computation method of H and A is as follows (Lempel and Moran 2001). The Web graph $G = (V, E)$ is used to construct a bipartite graph $G' = (V_h, V_a, E')$ where (Fig. 11.9):

- $V_h = \{s \mid s \in V, \text{outdegree}(s) > 0\}$, *hub side*;
- $V_a = \{s \mid s \in V, \text{indegree}(s) > 0\}$, *authority side*;
- $E' = \{(s, r) \mid (s, r) \in E\}$.

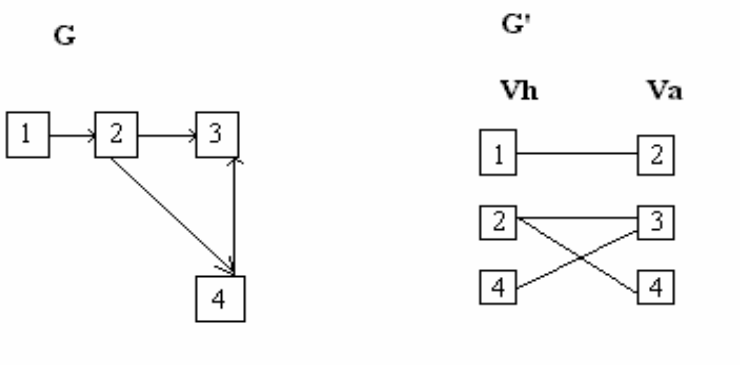


Fig. 11.9. Graph G, bipartite graph G' in the SALSA method

It was assumed that graph G was connected (if it is not connected, then the graph G' is constructed for every connected subgraph of G). Matrix $A = (a_{ij})$ was defined as follows:

$$a_{ij} = \sum_{\{k|(k,i),(k,j)\in E'\}} \frac{1}{\deg(i \in V_a)} \times \frac{1}{\deg(k \in V_h)}, \qquad (11.28)$$

while matrix $H = (h_{ij})$ was defined as follows:

$$h_{ij} = \sum_{\{k|(i,k),(j,k)\in E'\}} \frac{1}{\deg(i \in V_h)} \times \frac{1}{\deg(k \in V_a)}. \qquad (11.29)$$

11.6 The Associative Interaction Method

Before describing the method, the necessary notions and results from the theory of Artificial Neural Networks (ANN) are presented.

11.6.1 Artificial Neural Network

The fundamental principle of Artificial Neural Networks (ANNs), states that the amount of activity of any neuron depends on (James, 1890)

- its weighted input,
- the activity levels of artificial neurons connecting to it.

An *artificial neuron* is a formal processing unit abstracted from real, biological neurons (Fig. 11.10.a)) (Feldman and Ballard 1982, Grossberg 1976, Hopfield, 1984).

An artificial neuron ν has inputs $I_1,\ldots,I_n$. The inputs can be weighted by the weights $w_1,\ldots,w_n$. The total input I depends on inputs and their weights. The typical form of I is a linear combination of its inputs:

$$I = \sum_{i=1}^{n} I_i w_i \ . \qquad (11.30)$$

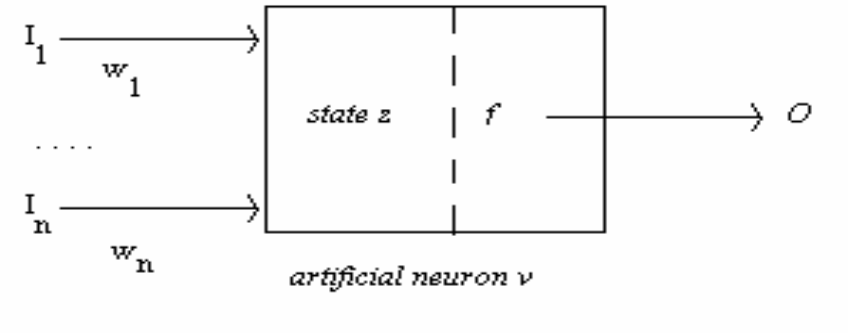
As a result of total input I, the neuron can take on a *state* (also called *activation level*) z. State z is a function g of I, $z = g(I)$. For example:

- threshold function: $z = 1$ if $I > k$, and $z = 0$ if $I \leq k$, where k is a threshold value;
- identity function: $z = I$.

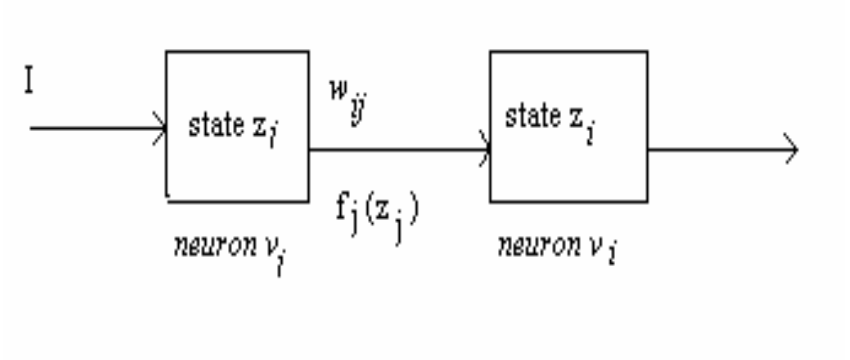
The artificial neuron produces an output O via its *transfer function* f depending on its state z, i.e., $O = f(z)$, for example:

- identity function: $O = f(z) = z$,
- sigmoid function: $O = f(z) = \dfrac{1}{1 + e^{-z}}$.

Artificial neurons can be connected to form an *Artificial Neural Network* (*ANN*; Fig. 11.10.b). Given two interconnected neurons ν_i and ν_j in an *ANN*, the *output* $f_j(z_j)$ of ν_j can be transferred to ν_i via the connection between them which can alter $f_j(z_j)$ by a weight w_{ij}. The quantity $w_{ij} \times f_j(z_j)$ reaches artificial neuron ν_i for which it is an input.



a) artificial neuron



b) artificial neural network

Fig. 11.10. **a)** An artificial neuron. A linear combination of the weighted (w_i) inputs (I_i) activates neuron *v* which takes on a state *z* and produces an output *O* via its transfer function *f*. **b)** ANN. Interconnected artificial neurons (ν_j, ν_i). *I* is an input to neuron ν_j, and $f_j(z_j)$ is its output, which is an input to neuron ν_i weighted by the quantity w_{ij}, i.e., $w_{ij} \cdot f_j(z_j)$

The state z_i of neuron ν_i can be described by the following generic differential equation (DeWilde, 1996):

$$\frac{dz_i(t)}{dt} = -z_i(t) + \sum_{j=1}^{n} f_j(w_{ij}, z_j(t), z_i(t)) + I_i(t), \qquad (11.31)$$

where

- t denotes time,
- $z_i(t)$ denotes the activity level of neuron ν_i,
- w_{ij} denotes the weight of a link from neuron ν_j to neuron ν_i,
- $I_i(t)$ denotes external input to neuron ν_i,
- $f_j(z_j(t),w_{ij},z_i(t))$ denotes the influence of neuron ν_j on neuron ν_i.

Eq. (11.31) is a generic equation, and can have different forms depending on the choice of I_i, f_j, w_{ij} corresponding to the particular case or application where the ANN is being used. For example, when applied to real (i.e., biological) neurons, then

- z_i denotes membrane voltage,
- I_i means an external input,
- w_{ij} is interpreted as a weight associated to the synapse,
- whereas f_j takes the form of a product between weight and z_j;

while for analogue electric circuits:

- z_i denotes the potential of a capacitor,
- the left hand side of the equation is interpreted as a current charging a capacitor to potential z_i,
- whereas the summed terms mean potentials weighted by conductance.

Because eq. (11.31) can be written for every i = 1, 2,...,n, we have a *system of differential equations*. The study of an *ANN* is carried out by assuming that *initial states* z_0 are known at some initial point t_0. It can be shown that in a small enough vicinity $|z–z_0|$ of z_0 and $|t–t_0|$ of t_0, system (11.31) has a unique solution. From a practical point of view, the existence of solutions eqs. (11.31) can be answered positively due

to the Cauchy-Lipschitz theorem (Martin and Reissner 1961), and is stated here without proof (as it is well-known in the theory of differential equations, on the other hand because it is the result of the theorem rather than the proof is important for us now in Information Retrieval):

Theorem 11.1. *Given the following system of differential equations:*

$$F(t,z) = \frac{1}{\mu_i}(I_i(t) - z_i(t) + \sum_j f_j(z_j(t), w_{ij}, z_i(t))),$$

where μ_i *is a coefficient. Consider the initial condition* $z(t_0) = t_0$*. If function* $F(t, z)$ *is continuous in a region* $\Omega \subset \mathbb{R}^2$ ($\mathbb{R}^2$ *denotes the real plane*)*, and function* $F(t, z)$ *is a local Lipschitz contraction, i.e.,*

$$\forall P \in \Omega \ \exists K \subset \Omega \text{ and } \exists L_K > 0 \text{ constant such that}$$
$$|F(t,z_1) - F(t,z_2)| \leq L_K |z_1 - z_2|, \ \forall (t,z_1), (t,z_2) \in K,$$

then there exists a vicinity $V_0 \subset \Omega$ *of point* (t_0, z_0) *in which the equation has a unique solution satisfying the initial condition* $z(t_0) = t_0$*, which can be obtained by successive numeric approximations.* □

Eqs. (11.31) give the state of every neuron at time *t*. By letting time *t* to evolve, a sequence $z_i(t)$, $i = 1,\ldots,n$, of states is obtained. This is referred to as the *operation* of *ANN*. Normally, an *ANN* evolves in time towards a state which does not change anymore. This is called an *equilibrium* and is given by the following equations:

$$\frac{dz_i}{dt} = 0, \quad i = 1, 2,...,n. \qquad (11.32)$$

An important mode of operation of an ANN is referred to as the *winner-take-all* (*WTA*) strategy which reads as follows: only the neuron with the highest state will have output above zero, all the others are 'suppressed'. In other words, *WTA* means selecting the neuron that has maximum state and deactivating all the others. Formally, the WTA can be expressed as follows:

$$(z_i = 1 \text{ if } z_i = \max_j z_j) \wedge (z_k = 0 \text{ if } z_k \neq \max_j z_j).$$

11.6.2 Associative Interaction Method

Let (Fig. 11.11):

- $\Delta = \{O_1, O_2,\ldots,O_i,\ldots,O_N\}$ denote a set of Web pages under focus. Each page O_i is assigned an artificial neuron $\aleph_i$, $i = 1,\ldots,N$. Thus, we may write: $\Delta = \{\aleph_1, \aleph_2,\ldots,\aleph_i,\ldots,\aleph_N\}$.
- $\Phi_i = \{\aleph_k \mid k = 1,\ldots,n_i\}$ denote the set of artificial neurons that are being influenced (i.e., synapsed, pointed to by) by $\aleph_i$, $\Phi_i \subseteq \Delta$.
- $B_i = \{\aleph_j \mid j = 1,\ldots,m_i\}$ denote the set of artificial neurons that influence (i.e., synapse to, point to) $\aleph_i$, $B_i \subseteq \Delta$.

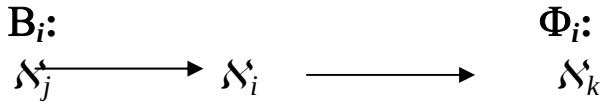


Fig. 11.11. $\aleph_1, \aleph_2,\ldots,\aleph_i,\ldots,\aleph_N$ form an artificial neural network. $\Phi_i=\{\aleph_k \mid k=1,\ldots,n_i\}$ denotes the set of artificial neurons that are being influenced by $\aleph_i$. $B_i=\{\aleph_j \mid j=1,\ldots,m_i\}$ denotes the set of artificial neurons that influence $\aleph_i$

The Associative Interaction Method is derived from the generic eq. (11.31) as follows (Dominich et al 2006).

Because the objects to be searched are Web pages, no external input (i.e., from outside the Web) can be assumed, so we take $I_i(t) = 0$.

One way to define f_j is to conceive the influence of a page j on another page i as being determined by the strengths of the connections which convey this influence, i.e., weights w_{ij} of the links between them. Eq. (11.31) thus reduces to the following equation:

$$\frac{dz_i(t)}{dt} = -z_i(t) + \sum_{\aleph_j \in \mathrm{B}_i} w_{ij} \tag{11.33}$$

In order to simplify the writing, let us introduce the following notation: $\sum_{\aleph_j \in \mathrm{B}_i} w_{ij} = \Sigma^{(i)}$. It is known from the theory of differential equations (Theorem 11.1) that the solution of eq. (11.33) has the following general form

$$z_i(t) = Ce^{-t} + \Sigma^{(i)}, \tag{11.34}$$

where C is a constant depending on the initial condition.

When the network operates for retrieval, activation spreading is taking place according to WTA strategy. At any time step t_u, $u = 0, 1,\ldots$, exactly one neuron $k \in \{1,\ldots,N\}$, i.e., the winner, is active, all the other neurons $s \in \{1,\ldots,k-1, k+1,\ldots,N\}$, $s \neq k$, are deactivated, i.e., $z_s(t_u) = 0$. Taking into account this initial condition, the activity level of any non-winner neuron s is as follows:

$$z_s(t) = (1 - e^{t_u - t})\Sigma^{(s)}. \tag{11.35}$$

If time t is let to increase, activity level $z_s(t)$ tends to stabilize on the total input value $\Sigma^{(s)}$ of that neuron s:

$$\lim_{t \to \infty} z_s(t) = \Sigma^{(s)}. \tag{11.36}$$

At the next time step t_{u+1}, of these neurons s, the winner will be the neuron p whose activity level z_p exceeds the activity level z_s of any other neuron s, i.e., $z_p \geq z_s$. This re-writes as follows:

$$(1 - e^{t_u - t})\Sigma^{(p)} \geq (1 - e^{t_u - t})\Sigma^{(s)}. \tag{11.37}$$

Because $t > t_u$, we have $e^{t_u - t} < 1$, and so $(1 - e^{t_u - t})$ is strictly positive. Hence, the winner-condition $z_p \geq z_s$ becomes equivalent to $\Sigma^{(p)} \geq \Sigma^{(s)}$. In other words, the neuron with the highest total input will be the winner.

Thus, from a **practical point of view, the Associative Interaction Method** can be applied in the following way.

Each Web page W_i is viewed as an artificial neuron, and is associated an n_i-tuple of weights corresponding to its terms (obtained after stemming and stoplisting) t_{ik}, $k = 1,\ldots,n_i$. Given now another page W_j. If term t_{jp}, $p = 1,\ldots,n_j$, occurs f_{ijp} times in W_i, then there is a link from W_i to W_j, and this may have the following weight (normalized frequency weighting scheme):

$$w_{ijp} = \frac{f_{ijp}}{\sum_k f_{ik}} . \tag{11.38}$$

If identifier t_{ik} occurs f_{ikj} times in W_j, and df_{ik} denotes the number of pages in which t_{ik} occurs, then there is another link from W_i to W_j, and this may have the following weight (inverse document frequency weighting scheme):

$$w_{ikj} = f_{ikj} \cdot \log\frac{2N}{df_{ik}} . \tag{11.39}$$

The total input to W_j is then

$$\sum_{k=1}^{n_i} w_{ikj} + \sum_{p=1}^{n_j} w_{ijp} . \tag{11.40}$$

The other two connections — in the opposite direction — have the same meaning as above:

- w_{jik} corresponds to w_{ijp},
- while w_{jpi} corresponds to w_{ikj}.

A query Q is considered to be a page, i.e., it is interlinked with pages (this process is referred to as *interaction* between query and pages) . The process of retrieval is as follows (Fig. 11.12).

- A spreading of activation takes place according to a WTA strategy.

- The activation is initiated at the query $Q = o_j$, and spreads over along the strongest total connection thus passing on to another page, and so on.
- After a finite number of steps, the spreading of activation reaches a page which has already been a winner earlier, this giving rise to a loop (referred to as a *reverberative circle*) This is analogous to a „local memory“ recalled by the query (this process may be conceived as a process of *association*: some pages are associated to the given query). Those pages are said to be retrieved, which belong to the same reverberative circle.

Fig. 11.13 shows example — and typical — plots of activity levels $z_s(t)$ for four neurons. It can be seen how activity levels reach asymptotically their limit which is equal to the corresponding total input values 1, 5, 3, 6.

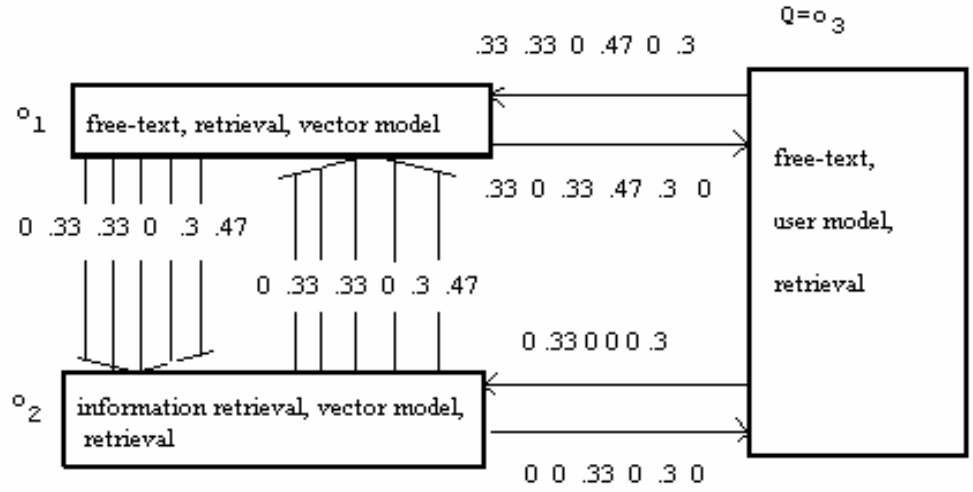


Fig. 11.12. Associative Interaction Retrieval Method (example). All links having the same direction between Q and o_1, and Q and o_3 are shown as one single arrow to simplify the drawing. The activation starts at *Q*, and spreads over to o_1 (total weight = .33 + .33 + .47 + .3 = 1.43) from which to o_2, and then back to o_1. o_1 and o_2 form a reverberative circle, and hence o_1 and o_2 will be retrieved in response to Q

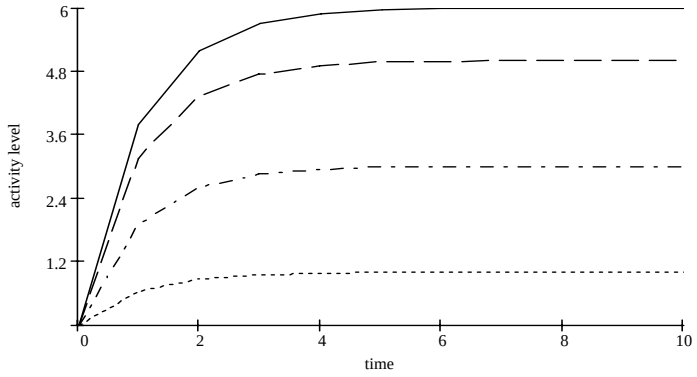


Fig. 11.13. Example plots of activity levels for four neurons in the Associative Interaction Method during the operation for retrieval. It can
be seen how activity levels reach asymptotically their limit which is equal to the corresponding total input. The highest will be the winner

11.6.3 Application of the Associative Interaction Method in Web Retrieval

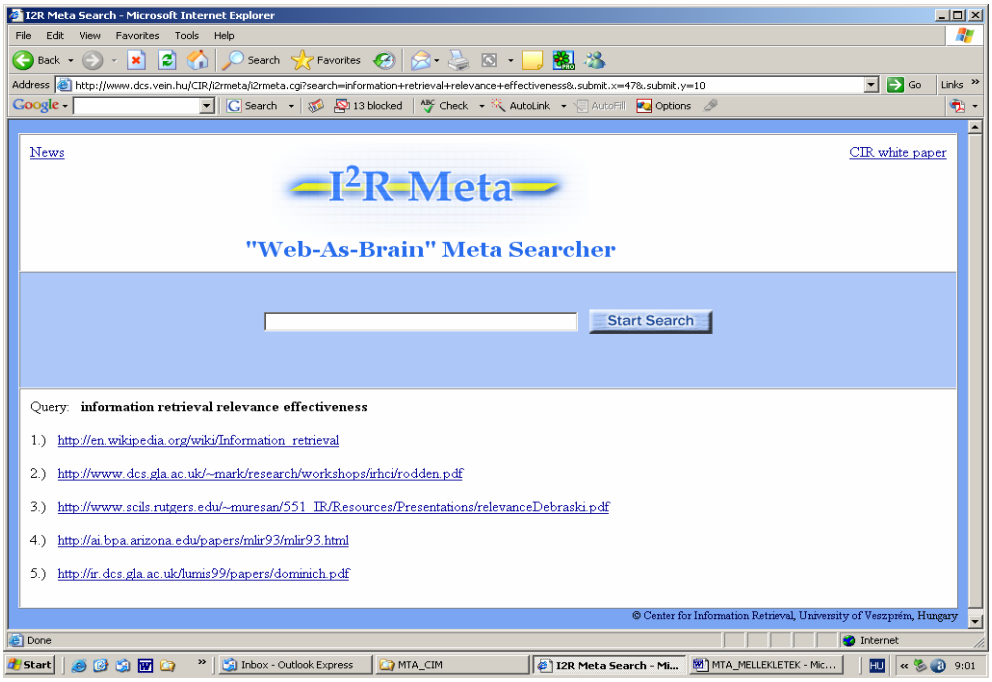


Fig. 11.14. Interface screen of the Web meta-search engine I^{2}RMeta

11.7 Combined Methods

Combined methods aim at computing the importance (used for ranking) of Web pages as a combination of

- their importance stemming from their belonging to a network of pages (link importance or link-based evidence),
- and of their importance given by their content (referred to as intrinsic importance or content-based evidence).

The Impact Factor, Connectivity, Mutual Citation, PageRank, HITS, SALSA, Associative-Interaction (or other) methods can be used to compute a link importance for a Web page.

11.7.1 Similarity Merge

It is worth recalling first an early method which combines link-based and content-based evidence as the sum of normalised deviations from minimum (Fox and Shaw 1994).

A combined importance Ψ_j for a Web page W_j under focus is computed using the following formula:

$$\Psi_j = \sum_{i=1}^{m} \frac{s_i - s_{\min}}{s_{\max} - s_{\min}},$$

where

- s_i is the importance of page W_j given by method i,
- $m > 1$is the number of methods used to calculate importance,
- $s_{min} = \min_i s_i$, $s_{max} = \max_i s_i$.

The combined importance Ψ_j can be weighted (i.e., multiplied) by a factor M denoting the number of methods for which $s_i \neq 0$.

11.7.4 Aggregated Method

Using the notions of fuzzy algebra and fuzzy probability (chapter 9), an aggregated method will be given for Web retrieval (mainly for retrieval on a given site).

11.7.4.1 Content-based Importance

A Web page W_j may be interpreted as being a fuzzy set in a set $T = \{t_1,\ldots,t_n\}$ of terms, i.e., $W_j = \{(t_i, \varphi_j(t_i)) \mid t_i \in T, i = 1,\ldots,n\}$. Then, its fuzzy probability P_j is

$$P_j = P(W_j) = \sum_{i=1}^{n} \varphi_j(t_i)\, p(t_i)\,, \qquad (11.41)$$

where $p(t_i)$ denotes a frequecny-based probability of term t_i. One way to compute it is as follows ("favorable cases over all cases"):

$$p(t_i) = \frac{\sum_{j=1}^{N} f_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{N} f_{ij}}\,, \; i = 1,\ldots,n, \qquad (11.42)$$

where

- n = number of terms,
- N = number of Web pages,
- $\varphi_j(t_i)$ = membership function (e.g., weight of term t_i for page W_j),
- f_{ij} = number of occurrences of term t_i in W_j.

P_j may be interpreted as being proportional to (or an indication of) the chance that the page is being selected or occurs (for example, in a hit list) based on its content. The fuzzy probability of a page is equal to zero if the page does not have any content (it is without meaning, the weights of its terms are all zero). The fuzzy probabilities $\Pi = [P_1,\ldots,P_j,\ldots,P_N]^T$ of all pages are given by the following matrix multiplication:

$$\Pi = \Phi \times P\,, \qquad (11.43)$$

where $\Phi = (\varphi_{ji})_{N\times n}$, $P = [p(t_1),\ldots,p(t_i),\ldots,p(t_n)]^{\mathsf{T}}$.

11.7.4.2 Combined Importance Function

A combined importance function Ψ of a Web page W is defined as being a function F

- of its link importance L (stemming from the link structure of the Web graph),
- and of its intrinsic importance given by the fuzzy probability P,

as follows:

$$\Psi = \Psi(P, L). \qquad (11.44)$$

From a practical point of view, an analytic form for the combined importance function Ψ should be given. In this regard, the following assumptions (or axioms) seem reasonable to be made.

Assumption 1. It seems straightforward to require that the combined importance of an isolated page without content be null:

$$\Psi(0, 0) = 0. \qquad (11.45)$$

Assumption 2. If a Web page does not carry any meaning (practically it does not have any content), i.e., $P = 0$, then its combined importance should vanish, even if it is highly linked; formally:

$$\Psi(L, 0) = 0, \quad L \neq 0. \qquad (11.46)$$

Note. This assumption may need further investigation, because, for example, a hub page may be very useful even if it contains only links.

Assumption 3. Further, from zero link importance ($L = 0$) need not necessarily follow a vanishing combined importance Ψ if the fuzzy probability does not vanish (e.g., this may be the case of a “young” Web page which is an isolated node of the Web graph, but which may carry important meaning). Formally:

$$\Psi(0, P) \neq 0, \ P \neq 0. \qquad (11.47)$$

Assumption 4. It seems natural to require that the combined importance of a page increase with its probability P for the same link importance L; the same should hold also for L. Formally:

$$P_1 < P_2 \Rightarrow F(L, P_1) < F(L, P_2),$$
$$L_1 < L_2 \Rightarrow F(L_1, P) < F(L_2, P). \qquad (11.48)$$

One possible and simple analytical form for Ψ that satisfies Assumptions 1-4 is as follows:

$$\Psi(L,P) = PL + aP = P(L + a), \qquad (11.49)$$

where parameter $a > 0$ is introduced to "maintain" a balance between the probability-based importance P and link-based importance L when P happens to be much larger than L. It can be easily seen that Ψ satisfies all of the Assumptions 1-4. Figure 11.15 shows the plot (surface) of the combined importance function Ψ defined by eq. (11.49).

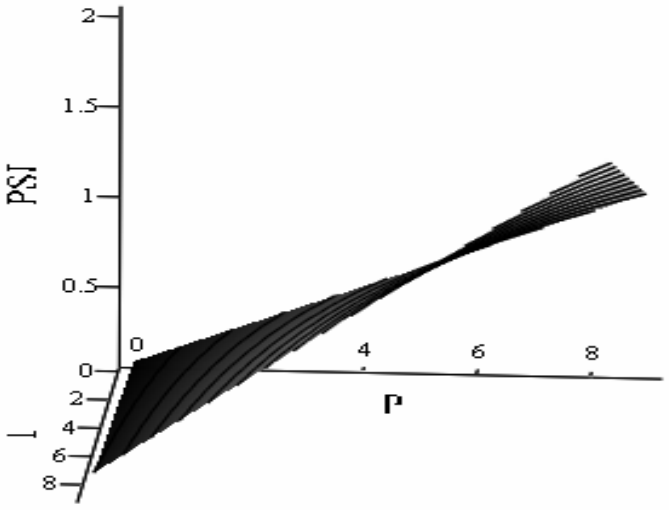


Figure 11.16. Graphical representation of the combined importance function Ψ = PSI = $PL+aP$ for $a = 1$ (The values on the L and P axes are grid points for scaling purposes, the L and P values are obtained by division by 10)

11.7.4.3 Aggregated Method

COMBINED METHOD

1. Construct the Web graph G for Web pages under focus, W_j, $j = 1,\ldots,N$.
2. Compute link importance L_j for every Web page under fcus, W_j, $j = 1,\ldots,N$. In principle, any method (Connectivity, PageRank, HITS, SALSA, Associative, or other) may be used. For example, using the PageRank method:

$$L_j = \alpha \cdot (1-d) + \beta \cdot d \cdot \sum_{W_k \in \mathrm{B}_j} \frac{L_k}{C_k} + \gamma \cdot E .$$

3. Construct a set of terms $T = \{t_1,\ldots,t_i,\ldots,t_n\}$.
4. Construct the term-page frequency matrix M:

$$M = (f_{ij})_{N \times n}.$$

5. Compute probabilities $p(t_i)$ as follows:

$$p(t_i) = \frac{\sum_{j=1}^{N} f_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{N} f_{ij}}$$

6. Define membership functions $\varphi_j(t_i)$, $j = 1,\ldots,N$; $i = 1,\ldots,n$. For example, $\varphi_j(t_i) = w_{ij}$, where the weight w_{ij} is calculated using a weighting scheme.
7. Calculate the fuzzy probability P_j of every Web page W_j, $j = 1,\ldots,N$,
as follows:

$$P_j = P(W_j) = \sum_{i=1}^{n} \varphi_j(t_i)\, p(t_i) .$$

8. Compute the combined importance Ψ_j for every Web page W_j (j = 1,…,N):

$$\Psi_j = L_j P_j + aP_j.$$

The Combined Method can be used to give the following Web retrieval method:

AGGREGATED RETRIEVAL METHOD

1. Given a query Q.
2. Compute similarities between Q and Web pages W_j (j=1,…,N):

$$\rho_j = \frac{\kappa(Q \cap W_j)}{P(Q)} = \frac{\sum_{i=1}^{n} q_i \cdot \varphi_j(t_i)}{\sum_{i=1}^{n} q_i \cdot p(t_i)} \cdot$$

3. Construct the set of pages which match the query as follows:

$$\{W_j \mid \rho_j \neq 0,\ j = 1,\ldots, J\}.$$

4. Compute an aggregated importance S_j for Web pages W_j (j=1,…,J) as follows:

$$S_j = \alpha\Psi_j + \beta\rho_j, \quad \alpha,\beta \text{ parameters.}$$

5. Rank pages $W_1,\ldots,W_J$ descendingly on their aggregated similarity $S_1,\ldots,S_J$ to obtain a hit list H.

6. Show the entire hit list H or a part of it (use cut off or threshold) to the user.