Discovering Web Document Associations for Web Site Summarization

K. Selçuk Candan[1]* and Wen-Syan Li[2]

[1] Computer Sci. and Eng. Dept, Arizona State University, Tempe, AZ 85287, USA
[2] CCRL, NEC USA, Inc., 110 Rio Robles M/S SJ100, San Jose, CA 95134, USA

Abstract. Complex web information structures prevent search engines from providing satisfactory context-sensitive retrieval. We see that in order to overcome this obstacle, it is essential to use techniques that recover the web authors' intentions and superimpose them with the users' retrieval contexts in summarizing web sites. Therefore, in this paper, we present a framework for discovering implicit associations among web documents for effective web site summarization. In the proposed framework, associations of web documents are induced by the web structure embedding them, as well as the contents of the documents and users' interests. We analyze the semantics of document associations and describe an algorithm which capture these semantics for enumerating and ranking possible document associations. We then use these asociations in creating context-sensitive summaries of web neighborhoods.

1 Introduction

Hypermedia has emerged as a primary means for storing and structuring information. This is primarily visible in continuously proliferating web based information infrastructures in corporate and e-commerce organizations. Yet, due to the continuously increasing size of these infrastructures, it is getting ever difficult for users to understand and navigate through such sites. Furthermore, these complex structures prevent search engines from providing satisfactory context-sensitive retrieval functionalities. We see that in order to overcome this obstacle, it is essential to use techniques that recover the web authors' intentions and superimpose it with the users' retrieval contexts. Therefore, in this paper, we present a framework for creating web site summarizations.

When an author or a web designer prepares a web document, he/she would put intended information not only on in the textual content of a page; but, also on the link structure of the web site. Thus, the content of web pages along with the structure of the web domain can be used to derive hints to summarize web sites. What differentiates our work from similar work in the literature is that, we propose to use *associations between documents in a neighborhood and the reasons why they are associated* in the summarization process. Knowing these

* This work was performed when the author visited NEC, CCRL.

reasons, among other things, is essential in (1) in creating web site maps that matches web designers' intentions and (2) in superimposing the logical structure of a web site with the context provided by a visitors interests.

In the next section, we describe how to construct of meaningful summarizations of web neighborhoods, which can be viewed and used as a web site map. In Section 3, we present how to mine web document associations using page content as well as link structures. In 4, we describe the experiment results for our web site summarization algorithms, respectively. Finally we conclude this paper with discussion of future work.

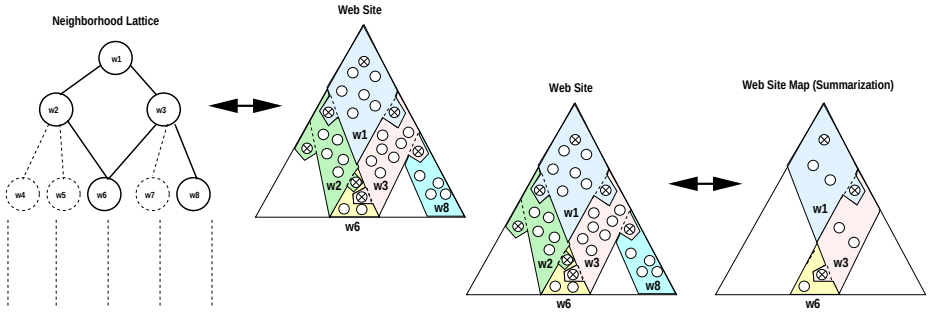
2 Web Site Summarization

In this section, we introduce the web site summarization task and discuss how to use document associations for this purpose. We see that corporate sites, and most of the web space, is composed of two types of neighborhood: physical and logical. The physical neighborhoods are separated from each other through domain boundaries or directory names. The logical neighborhoods, on the other hand, are mostly overlapping and they can cover multiple domains or they can be limited to a part of a single domain. Web designers usually aim at overlapping the physical neighborhoods of the web site with the logical neighborhoods. However, as (1) the foci of users may differ from each other, (2) the focus of a single user may shift from time to time, such a strict design may loose its intended effect over time.

Physical neighborhoods are usually decided by the URL structures of the web sites. In [1], we described algorithms to discover logical neighborhoods. Consequently, in this section, we will assume that a corporate web site, W, is already partitioned into its neighborhoods. We denote this partitioning as a partially ordered lattice $\mathcal{W}$:

- each vertex w_i in $\mathcal{W}$ is a set of nodes (or a neighborhood) and
- if w_j is a child of w_i, then w_j is a sub-neighborhood of w_i; furthermore, $w_i \cap w_j = \{v_{ij}\}$, where v_{ij} is the entry point to sub-neighborhood w_j from neighborhood w_i (Figure 1(a)).

Intuitively, the lattice corresponds to a hierarchy of neighborhoods. At the highest level, we have a neighborhood which consists of high-level corporate pages and the entry pages of lower neighborhoods. Similarly, each neighborhood consists of a set of high-level pages and the entry-pages of all its sub-neighborhoods. Consequently, summarization of W involves of two tasks: (1) identification of which nodes in the partially ordered lattice $\mathcal{W}$ will be shown to the user (i.e., focusing on the neighborhoods) and (2) summarization of each focussed neighborhood based on user interest (Figure 1(b)). Below, we discuss these to tasks in greater detail:



(a) Entry points (b) Summarization

Fig. 1. The crossed circles denote the entry points of neighborhoods. Each sub-neighborhood contain one entry point per its parent neighborhoods. Each parent neighborhood includes the entry points of its sub-neighborhoods.

2.1 Identification of Focus Neighborhoods

In order to identify the focus neighborhoods, we need to start from the root neighborhood, w_1 of $\mathcal{W}$. This neighborhood contains, the high-level pages of the given corporate site along with the entry level pages of it sub-neighborhoods. Let us assume that we are interested in a predetermined number, k, of focus points in this top neighborhood. Then, we could rephrase the focus identification problem in terms of the neighborhood summarization problem:

- In order to identify the k focal points of the neighborhood w_1 of $\mathcal{W}$, summarize w_1 into a graph of size k. The k remaining pages are in user focus.
- Let us assume that $F = \{v_1, v_2, \ldots, v_k\}$ are the k pages in the summary of w_1. If $v_i \in F$ is an entry-page of a sub-neighborhood, w_i, then repeat the same process for the sub-neighborhood w_i.

Note that the above recursive process allow us to identify the focal points of a web site. The parameter k is user-dependent and describes how focussed the user would like to be: smaller values of k correspond to more focussed site maps. Note also that, this recursive algorithm assumes that given a neighborhood, we can create a summary of size k. Next, we describe how we can achieve this task.

2.2 Summarization of a Neighborhood

Each neighborhood, $w_i \in \mathcal{W}$ consists of a set of neighborhood pages and the entry-pages of its sub-neighborhoods. Also, from each of its parent-neighborhoods, w_i is reached through one entry-page (Figure 1). Let us assume that the set, $\mathcal{E}$, of pages correspond to the entry pages of w_i from all its parent-neighborhoods that are in focus. Then our goal is to summarize the neighborhood with respect to these entry points as well as the content-description provided by the user. The summarized neighborhood will give the focussed pages in this neighborhood and the corresponding connectivity.

In order to summarize a given neighborhood, we first have to identify the pages that are important. In this case, the entry pages of a neighborhood (from parents in focus) are relatively important as they will connect the web site maps of the neighborhoods. Note also that the entry pages of the sub-neighborhoods are also important as they will extent the map downwards in the hierarchy, given that the lower neighborhoods are also in focus.

Therefore, given a neighborhood, w_i, the set, $\mathcal{E}$, of focussed entry pages from its parents, and the set, $\mathcal{L}$, of entry-pages to its sub-neighborhoods, we can create a set of seed pages (for summary) $\mathcal{S} = \mathcal{E} \cup \mathcal{L}$. Then our goal is,

- given the set, $\mathcal{S}$, of seed (entry) web pages,
- potentially a content-description,
- a web neighborhood, $G^N = w_i$ which contains these seeds, and
- an integer k,

to create a *summary*, with k pages, of the neighborhood with respect to the seed pages.

Observation: Since, the web site map is a set of representative nodes in a web site, the nodes in a web site map needs to satisfy the following criteria:

- High connectivity so that users can navigate from these web site map nodes to other nodes easily.
- The contents of these web site map nodes need to be representative.

Thus, the nodes selected to form a web site map need to be both structural and content-wise representative for all pages in a web site. Therefore, we can use the selected nodes, *which describe the association between the pages in the site,* as a summary of a web site and to form a site map.

Therefore, given a graph $G(V, E)$, its undirected version $G^u(V, E^u)$, and k *dominant vertices* in V with respect to the given seed vertices $\mathcal{S}$, we can construct a k-summary $\Sigma^k_{(\mathcal{S})}(V^\sigma, E^\sigma, \delta)$ of the input graph (δ is a mapping, $E^\sigma \rightarrow R^+ \times \{left, right, bi, none\}$, which describes the lengths and directions of the summarized edges) as shown in Figure 2.

Note that Step 3 of the algorithm requires the identification of the k most dominant vertices (or the vertices which describe the document associations the best) in the graph with respect to the seed vertices. These vertices will be the only vertices used in the summarization (Step 4). Given two dominant vertices, v_i and v_j, (to be visualized in the summary) Step 5 of the algorithm first constructs a temporary graph, $G_{temp}(V_{temp}, E_{temp})$, from the original web graph, such that no path between v_i and v_j can pass through another dominant node. Then, it uses the shortest path, $sp(v_i, v_j)$, between v_i and v_j in this temporary graph to identify edges that are to be visualized to the user. Consequently, a given edge that is included in the summary denote the shortest path, between two dominant vertices, that do not pass through other dominant vertices. Hence, this step eliminates the possibility of inclusion of redundant edges.

1. Let $G^N(V^N, E^N) \subseteq G^u$ be the neighborhood graph;
2. $V^\sigma = \emptyset$; $E^\sigma = \emptyset$;
3. Let $\mathcal{K} \subseteq V_{G^N}$ be the set of k vertices with the highest dominance values;
4. $V^\sigma = \mathcal{K}$;
5. For each v_i and $v_j \in V^\sigma$
   (a) $V_{temp} = V^N - \{V^\sigma - \{v_i, v_j\}\}$;
   (b) $E_{temp} = \{\langle v_k, v_l\rangle \mid (\langle v_k, v_l\rangle \in E^N) \wedge (v_k \in V_{temp}) \wedge (v_l \in V_{temp})\}$
   (c) If $sp(v_i, v_j)$ is the shortest path in $G_{temp}(V_{temp}, E_{temp})$ between v_i and v_j then
      i. Let the length of the path $sp(v_i, v_j)$ be Δ
      ii. $E^\sigma = E^\sigma \cup \{e = \langle v_i, v_j\rangle\}$;
      iii. If v_j is reachable from v_i in the directed graph G through the vertices in $sp(v_i, v_j)$, but if v_i is not reachable from v_j, then $\delta(e) = \langle \Delta, right\rangle$
      iv. If v_i is reachable from v_j in the directed graph G through the vertices in $sp(v_i, v_j)$, but v_j is not reachable from v_i, then $\delta(e) = \langle \Delta, left\rangle$
      v. If v_i and v_j are reachable from each other in the directed graph G through the vertices in $sp(v_i, v_j)$, then $\delta(e) = \langle \Delta, bi\rangle$
      vi. If neither v_i nor v_j is reachable from the other in the directed graph G through the vertices in $sp(v_i, v_j)$, then $\delta(e) = \langle \Delta, none\rangle$

Fig. 2. Algorithm for constructing a summary

Note that, for the identification of shortest paths to be visualized, the path length can be defined in various ways. One possibility would be *minimization of the number of edges on the path.* This would be useful when the aim is to display the user information about the connectivity of the summarized graph. Once the edges to include in the summary, the sub-steps of Step 5(c) gathers more information regarding each edge (such as the reachability of the two pages at the end-points from each other in the inherently directed web) and reflects this information to the summary in the form of edge labels.

Example 1. Let us consider the graph in Figure 3(a). and let us assume that we are asked to construct 5- and 7-summaries of it. For this example, let us also assume that after running an association mining algorithm, we have found that the seven dominant vertices in this neighborhood (relative to a set of seed vertices) are A, B, F, E, J, C, and I. For simplicity, let us also assume that the length of the path is defined using the number of edges on it. Figures 3(b) and (c) shows the corresponding 5- and 7-summaries of this graph. The labels of the edges denote the length of the corresponding paths and the arrows on the edges denote the reachability of the end-points from each other (e.g., an edge of the form "$\longleftrightarrow$" denotes that both end-points can reach the other one over the web through this shortest path; whereas, an edge of the form "—" denotes that neither of the end-points can reach the other one over the web through this shortest path.

Those entry pages which are still in the map after the summarization are called *focussed entry-pages*, and they point to the other logical domains that have

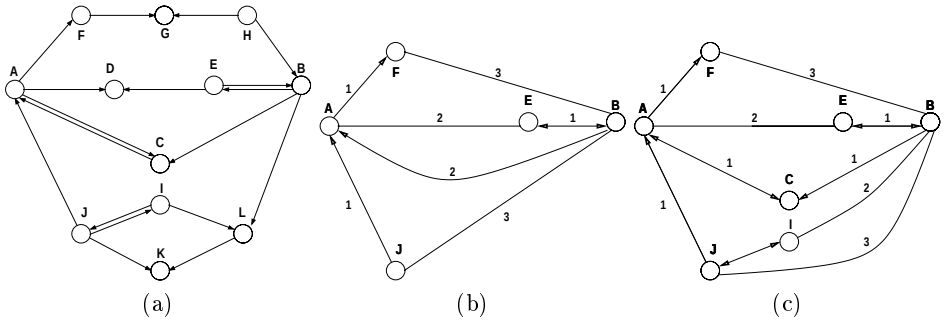


Fig. 3. (a) The web neighborhood in our running example, (b) its web site map with 5 nodes, and (c) its web site map with 7 nodes

to be further explored and summarized. Therefore, we recursively apply the summarization algorithm described above for those domains who have at least one *focused entry-page*.

3 Finding Document Associations

We now use an example to illustrate the task of idenifying document associations and overview of our approach.

Example 2. In Figure 4, we show a set of links between two web pages `W.Li` and `D.Agrawal` (both are highlighted in the figure). The purpose of each web link is indicated as its label in the figure. For example, `W.Li` graduated from Northwestern University; whereas `D.Agrawal` and `P.Scheuermann` both graduated from SUNY. Based on this link structure, if we want to find the association between `W.Li` and `D.Agrawal` pages, the link structure connecting `W.Li` and `D.Agrawal` are useful clue for mining. Below, we enumerate some associations that are embedded in this graph:

- *Association 1:* `Web8 paper` page appears in a path of distance of 2 connecting the pages `W.Li` and `D.Agrawal`. Therefore, `W.Li` and `D.Agrawal` may be associated due to a co-authored paper.
- *Association 2:* `Y.Wu` page is on two paths related to NEC Research Laboratories, each of distance 4. `W.Li` and `D.Agrawal` may be associated due to the fact they both supervised `Y.Wu` at different occasions or they participate in the same project at NEC.
- *Association 3:* `WOWS'99` and `ACM DL'99` pages appear on a single path of distance 3. Such an association can be interpreted as that `W.Li` and `D.Agrawal` are participating in the same conference (e.g. presentation or program committee members).

The above example shows that the following two intuitions, along with the actual content of the pages, can generally be used to identify why a given set of pages are associated:

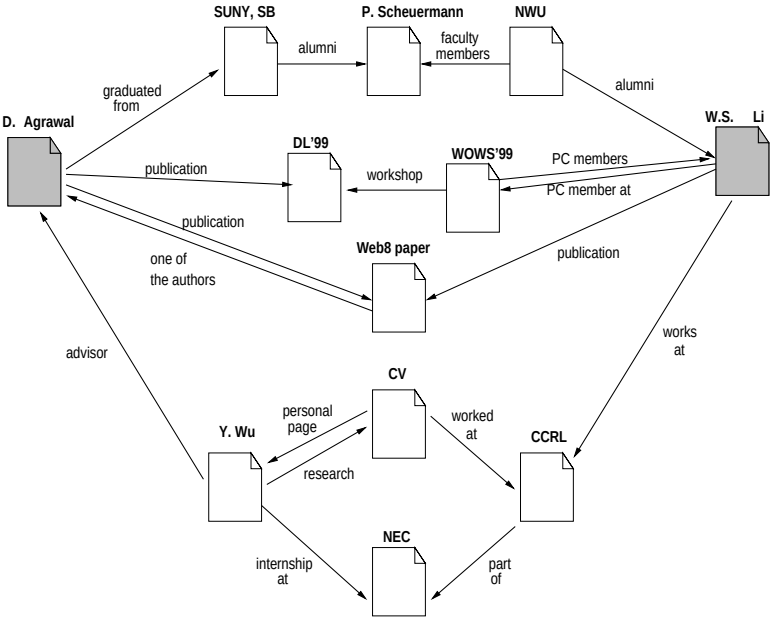


Fig. 4. Link structure associating the web documents `W.Li` and `D.Agrawal`

1. Pages on a shorter path between the `W.Li` and `D.Agrawal` pages are stronger indicators than others to reflect why the `W.Li` and `D.Agrawal` pages are associated.
2. Pages which appear on more paths should be stronger indicators than others to reflect why the `W.Li` and `D.Agrawal` pages are associated.

Note that a page with a higher connectivity (i.e. more incoming links and outgoing links) is more likely to be included in more paths; consequently, such a page is more likely to be ranked higher according to the above criteria. This is consistent with the principle of topic distillation[2, 3]. On the other hand, a page with a high connectivity but far away from the seed URLs may be less significant to represent the associations than a page with low connectivity but close to the seed URLs. A page which satisfies both of the above criteria (i.e. near seed URLs and with high connectivity) would be a good representative for the association.

Obviously, the *distance* between two pages can be defined in various ways. In the simplest case, the number of links between two pages can be used as the distance metric. On the other hand, in order to capture the physical as well

as logical distances between pages, we use different distance metrics capable of capturing document contents as well as user interests.

Based on this motivation, we use a novel framework for mining associations among web documents using information *implicitly* reflected in the links connecting them, as well as *the contents* of these connecting documents. We develop a web mining technique, based on a *random walk algorithm*, which considers three factors: (1) document distances by link; (2) connectivity; and (3) document content. Consequently, instead of explicitly defining a metric, we choose a set of random walk parameters that will implicitly capture the essence of these observations. The details and the complexity of this algorithm has been analytically and experimentally studied in [4]. In the next section, we present the experimental results on Web site summarization.

3.1 Comparison with Other Approaches

Note that the well-known algorithm *topic distillation* [2, 5, 6] could be a natural choice for summarization purposes. However, we observe that the behavior of the topic distillation algorithm may not be as good as our *document association* based approach in the scope of summarization tasks. The reason is that the topic distillation algorithm aims at selecting a small subset of the most "authoritative" pages and "hub" pages from a much larger set of query result pages. An authoritative page is a page with many incoming links and a hub page is a page with many outgoing links. Such authoritative pages and hub pages are mutually reinforcing: good authoritative pages are linked by a large number of good hub pages and vice versa. Because the important pages are mutually reinforcing, the results tend to form a cluster. For example, there are many on-line HTML papers at `www-db.stanford.edu`. These papers consist of a number of pages linking to each other. By using the topic distillation algorithm, most of these pages are selected while many individual home pages are left out; this is not suitable for the purposes of summarization. On the other hand, our algorithm uses the concept of *seed nodes* which focus the summarization process, based on the web site structure as well as the user focus. Nodes selected after summarization are those nodes that explain why the seed nodes are related. Therefore a good choice of seed nodes (in our case, the entry-nodes of logical domains) leads into a good and meaningful summarization. Thus, a document association based approach is more suitable for the purpose of summarization. Researchers in the AI community have developed Web navigation tour guides, such as WebWatcher[7]. WebWatcher utilizes user access patterns in a particular Web site to recommend users proper navigation paths for a given topic. User access paterns can be incorporated into the random walk-based algorithm to improve the document association mining.

4 Experiments on Web Site Map Generation

We have conducted a set of experiments on *www-db.standford.edu*, which has 3040 pages and 12,581 edges. The average number of edges per pages is 3.5. The

Table 1. Summarization results: (a) root domain and (b) a subdomain

Score	URL
0.124	/LIC/LIC.html
0.110	/LIC/mediator.html
0.032	/people/
0.031	/~gio/
0.018	/~wangz/
0.018	/~jan/watch/intro.html
0.016	/tsimmis/
0.016	/~danliu/
0.015	/cs347/
0.015	/LIC/
0.015	/~widom/
0.014	/~wilburt/
0.014	/~chenli/
0.013	/~ullman/
0.012	/CHAIMS/
0.012	/~echang/
0.012	/~cyw/
0.012	/~crespo/
0.012	/~cho/
0.012	/~sergey/

(a)

Score	URL
0.015	/pub/gio/CS545/image.html
0.011	/pub/gio/1999/Interopdocfigs.html
0.011	/pub/gio/biblio/master.html
0.011	/pub/gio/CS99I/library.html
0.010	/pub/gio/1994/vocabulary.html
0.009	/pub/gio/CS99I/ubi.html
0.009	/pub/gio/CS99I/entedu.html
0.009	/pub/gio/CS99I/health.html
0.008	/pub/gio/CS99I/wais.html
0.007	/pub/gio/CS99I/security.html
0.006	/pub/gio/CS99I/refs.html
0.005	/pub/gio/gio-papers.html
0.005	/pub/gio/inprogress.html
0.005	/pub/gio/
0.003	/pub/gio/paperlist.html
0.003	/pub/gio/CS99I/description.html
0.003	/pub/gio/CS99I/background/Cairn.....
0.003	/pub/gio/CS545/
0.003	/pub/gio/CS99I/copyright.html
0.002	/pub/gio/CS545/indexing/

(b)

experiments were ran on a 500MHz Pentium Architecture Linux OS PC with 128 MB of RAM. Using this setup, we have conducted a set of experiments to validate the web site summarization algorithm presented in this section. Here, we report on the main findings using one example case:

We asked our system to summarize the *www-db.standford.edu* domain, with respect to a context defined as "publication or paper". We also asked our system to give higher importance to more recently updated pages. First of all, our system identified 42 logical domains among 3040 pages. Then, we used the algorithm described in this paper, to recursively summarize this logical structure with respect to the defined context.

The algorithm started by summarizing the root logical domain which consists of all the entry-pages of the logical domain in the second level and the all the page in root logical domain, where *www-db.stanford.edu* is the entry page. Thus, 1584 pages are included for the experiments. The result of this summarization, using a radius of 2, is shown in Table 1(a) (we omit the summarized edge to simplify the discussion). Note that most of these pages are actual home pages and entry pages to the lower level logical domains. This was due to the fact that only these home pages are relevant to the focused keywords *papers* and *publications* and their edges are assigned with a lower cost. Thus, the algorithm prefers to "walk" through these pages over other 1400 pages. Some pages, such as

`www-db.stanford.edu/LIC/` and `www-db.stanford.edu/LIC/mediator.html`, in the root logical domain are included due to its high connectivity.

Next, the algorithm recursively visited the nodes in this domain. Here we report on one of the largest subdomains, i.e., *http://www-db.stanford.edu/~gio/*, (793 pages). When summarized with the proposed algorithm, the resulting graph contained the pages shown in Table 1(b). Note that these pages are either publication oriented pages, or they are linked to many pages with publication content: actual publication pages are omitted from the summarization to give place to pages which connect to many publication pages, allowing easier access to more information while browsing the web.

5 Conclusion

Hypermedia has emerged as primary means for structuring documents and for accessing the web. In this paper, we present a framework for site map construction and web page summarization. For this purpose, we introduce a random walk algorithm for mining implicit associations among web documents, induced by Web link structures and document contents. Link information has been used by many search engines to rank query results as well as finding relevant documents and web sites. We compare and contrast our random walk algorithm with other existing work, such as various topic distillation techniques.

References

[1] Wen-Syan Li, Okan Kolak, Quoc Vu, and Hajime Takano. Defining Logical Domains in a Web Site. In *Proceedings of the 11th ACM Conference on Hypertext*, pages 123–132, San Antonio, TX, USA, May 2000.

[2] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[3] Wen-Syan Li and K Selçuk Candan. Integrating Content Search with Structure Analysis for Hypermedia Retrieval and Management. *ACM Computing Surveys*, 31(4es):13, 1999.

[4] K. Selçuk Candan and Wen-Syan Li. Using Random Walks for Mining Web Document Associations. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 294–305, Kyoto, Japan, April 2000.

[5] Krishna Bharat and Monika Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21th Annual International ACM SIGIR Conference*, pages 104–111, Melbourne, Australia, August 1998.

[6] Lawrence Page and Sergey Brin. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th World-Wide Web Conference*, Brisbane, Queensland, Australia, April 1998.

[7] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *Proceedings of the 1997 Internaltional Joint Conference on Artificial Intelligence*, August 1997.