Mining Web Site Link Structures for Adaptive Web Site Navigation and Search

Jianhan Zhu BSc

Faculty of Informatics
University of Ulster at Jordanstown

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

October, 2003

University of Ulster

Abstract

Mining Web Site Link Structures for Adaptive Web Site Navigation and Search

by Jianhan Zhu

Supervisors:

Dr. Jun Hong
Faculty of Informatics

Professor John G. Hughes
Faculty of Informatics

This thesis is concerned with mining the log file of a Web site for knowledge about the Web site and its users, and using the knowledge to assist users to navigate and search the Web site effectively and efficiently. First, we investigate approaches to adapting the organization and presentation of a Web site by learning from the Web site link structure and user behavior of the Web site. Approaches are developed for presenting a Web site using a link hierarchy and a conceptual link hierarchy respectively based on how users have used the Web site link structure. Link hierarchies and conceptual link hierarchies can be used to help users navigate the Web site. Second, we develop approaches for building a first-order Markov chain model of user navigation on the Web site link structure, link hierarchy, and conceptual link hierarchy respectively. Under a collaborative assumption, the model can be used for link prediction that assists users to navigate the Web site. Third, approaches are developed for ranking Web pages based on how users have used the Web site link structure. The page rankings can be used to help users search the Web site.

The approaches developed in the thesis have been implemented in a prototype called *Online Navigation Explorer* (ONE). First, link hierarchies and conceptual link hierarchies are visualized in ONE. Second, link prediction using Markov chain models is integrated with link hierarchies and conceptual link hierarchies in ONE. Third, search results are visualized in ONE. Experimental results show that ONE can help users navigate a Web site and search for their desired information on the Web site effectively and efficiently.

The work presented in the thesis is a step towards the development of an adaptive Web site, which can assist users to navigate the Web site and search for their desired information on the Web site.

TABLE OF CONTENTS

STATEMENT

I hereby declare that the work reported in this thesis is the result of my own independent investigation unless otherwise stated. This work has not been, nor is it currently being, submitted in consideration for any other degree.

PhD Candidate: Jianhan Zhu

Signature: ______________________

Date: ______________________

ACKNOWLEDGEMENTS

NOTE ON ACCESS TO CONTENTS

I hereby declare that with effect from the date on which the thesis is deposited in the Library of the University of Ulster, I permit the Librarian of the University to allow the thesis to be copied in whole or in part without reference to me on the understanding that such authority applies to the provision of single copies made for study purposes or for inclusion within the stock of another library.*This restriction does not apply to the British Library Thesis Service (which is permitted to copy the thesis on demand for loan or sale under the terms of a separate agreement) nor to the copying or publication of the title and abstract of the thesis* IT IS A CONDITION OF USE OF THIS THESIS THAT ANYONE WHO CONSULTS IT MUST RECOGNIZE THAT THE COPYRIGHT RESTS WITH THE AUTHOR AND NO QUOTATION FROM THE THESIS AND NO INFORMATION DERIVED FROM IT MAY BE PUBLISHED UNLESS THE SOURCE IS PROPERLY ACKNOWLEDGED.

Signature: _____________________

Date: _____________________

PREFACE

This thesis is the final outcome of my Ph.D. study in the School of Computing and Mathematics, Faculty of Informatics, University of Ulster at Jordanstown. It serves as documentation of my research work, which has been done between May 1999 and May 2003. My study has been supported by a Vice Chancellor Research Scholarship of the University of Ulster. The thesis consists of seven chapters.

Chapter 3 is based on the following paper:

Zhu, J., Hong, J. and Hughes, J. G. (2003) PageCluster: Mining Conceptual Link Hierarchies from Web Log Files for Adaptive Web Site Navigation. *ACM Transactions on Internet Technology (ACM TOIT)*, in press, 26 pages.

Chapter 4 is based on the following papers:

Zhu, J., Hong, J. and Hughes, J. G. (2002a) Using Markov Chains for Link Prediction in Adaptive Web Sites. In *Proc. of Soft-Ware 2002: the First International Conference on Computing in an Imperfect World*, pp. 60-73, Lecture Notes in Computer Science, Springer, Belfast, April.

Zhu, J., Hong, J. and Hughes, J. G. (2002b) Using Markov Models for Web Site Link Prediction. In *Proc. of the Thirteenth ACM conference on Hypertext and Hypermedia (Hypertext'02)*, pp. 169-170, ACM Press, College Park, MD, USA, June 11-15.

Zhu, J. (2001) Using Markov Chains for Structural Link Prediction in Adaptive Web Sites. In *Proc. of the 8th International Conference on User Modeling*, Lecture Notes in Computer Science 2109 Springer 2001, ISBN 3-540-42325-7: pp. 298-300, Sonthofen, Germany, July.

Chapter 5 is based on the following paper:

Zhu, J., Hong, J. and Hughes, J. G. (2001) PageRate: Counting Web users' votes. In *Proc. of the Twelfth ACM conference on Hypertext and Hypermedia (Hypertext'01)*, pp. 131-132, ACM Press, Århus, Denmark, August.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The two predominant paradigms for finding information on the Web are navigation and search [Olston and Chi 2003]. Most Web users typically use a Web browser to navigate a Web site. They start with the home page or a Web page found through a search engine or linked from another Web site, and then follow the hyperlinks they think relevant in the starting page and the subsequent pages, until they have found the desired information in one or more pages. They may also use search facilities provided on the Web site to speed up information searching. For a Web site consisting of a very large number of Web pages and hyperlinks between them, these methods are not sufficient for users to find the desired information effectively and efficiently.

On the other hand, contents of Web pages, extended anchor texts[1], hyperlinks between Web pages, and Web usage data of a Web site are rich sources of data for mining knowledge about the Web site and its users. The knowledge can be used to assist users to navigate the Web site and search for desired information more effectively and efficiently. By viewing Web pages as nodes and hyperlinks between them as directed edges between nodes, we can construct a link structure of a Web site. Contents of the Web pages and extended anchor texts are properties of these nodes and hyperlinks. Hyperlinks convey conceptual relationships between Web pages. User traversals on hyperlinks can be extracted from Web usage data and used as properties of the hyperlinks. The constructed *link structure*[2] therefore contains information about Web site contents, hyperlinks, and user behavior. This thesis aims to employ machine

[1] An extended anchor text consists of the anchor text itself and the text surrounding it.
[2] The link structure of a Web site may sometimes refer to the topology of a Web site only.

learning and data mining techniques in mining the Web site link structure for designing an *adaptive Web site* [Perkowitz and Etzioni 1998], which can automatically change its presentation and organization to assist user navigation and search by learning from Web site link structures.

1.1 Important Issues That This Thesis Addresses

Relying solely on Web browsers to navigate a Web site consisting of a large number of Web pages and hyperlinks between them has created some navigation problems for users, i.e., it is not always easy for users to find the desired information on the Web site effectively and efficiently. When they get frustrated in their attempts to find the desired information, it only takes them one click to leave the Web site. Thus how to help users navigate a Web site and find the desired information effectively and efficiently is crucial to the success of the Web site.

Nielsen [2000] identified three fundamental navigation questions that users might ask when they navigate a Web site, namely,*Where am I now? Where have I been? Where can I go next?* Current Web browsers cannot provide users with satisfactory answers to these questions. Nielsen emphasized the importance of structural navigation in the new generation of Web browsers. The Web site link structure should be visualized into different levels for users to know their current locations relative to the Web site as a whole. The visualized Web site link structure can help users understand the relationships between the Web pages they have visited. Users can decide where they can go next by knowing their current locations and the Web pages they have visited in the context of the Web site link structure. By visualizing the Web site link structure, it has been made easier for users to answer the above three navigation questions. Most Web sites may have an underlying *hierarchical organization*, which is convenient for both Web site designers to organize information and users to navigate among the Web pages that have been divided onto different conceptual levels [Rosenfeld and Morville 1998; Farkas and Farkas 2000; Nielsen 2000]. However, the hierarchy is often buried in a maze-like link structure, which is updated frequently.

Most Web sites are designed with a one-size-fits-all philosophy: the site designer determines the needs of the users and builds the link structure accordingly. However, one size frequently does not fit all. Users may use the site in ways that are different from the designer's expectations and view the Web site link structure differently from the designer [Anderson 2002].

- **Visualizing Web Site Link Structures for User Navigation**

The first issue that this thesis addresses is to visualize a Web site in a hierarchy in order to answer the first two navigation questions, namely, Where am I now? Where have I been? The hierarchy can be constructed based on the link structure of the Web site and how users have used the Web site link structure. In the constructed hierarchy, the Web site link structure created by the designer has been adapted to the views of users in using the Web site link structure, and can thus reflect users' views of the Web site. The hierarchy can help users understand the relationships between the Web pages they have visited. Using the hierarchy, users can control their navigation and thus are not confined to following only the hyperlinks in each page. They can understand their current locations in the context of different levels of pages in the hierarchy. Users can decide where they can go next by understanding their current locations and relationships between the pages they have visited. The hierarchy can be seen as a form of adaptation of the Web site link structure to user behavior.

- **Link Prediction for User Navigation**

The second issue that this thesis addresses is how to predict a user's Web navigation given the pages the user has visited on the Web site in order to answer the third navigation question, namely, where can I go next? We address this issue by developing a two-step process of modeling and prediction. We use a *collaborative approach* for doing this, which assumes that a user typically behaves in a similar way as other users. A user model can be built using data from a group of users, and it is then used to make predictions about a new user in the absence of information about him/her. When a user visits a Web site, a user model built with the collaborative approach will use its information about all the past users of the site in order to predict the pages the user is most likely to request next [Zukerman et al. 1999]. Link prediction can be integrated with visualized hierarchy to answer all the three navigation questions.

Olston and Chi [2002] argued that the two predominant paradigms for finding information on the Web are navigation and keyword-based search. Keyword-based search is popular for quickly identifying pages containing specific information. On the other hand, navigation is useful when keyword-based search is difficult to be carried out for a variety of reasons, e.g., the user may not be certain of what he/she is looking for until the available options are presented, and some complex information-searching tasks are hard to be formulated in keywords. While they exhibit complementary advantages, neither paradigm is adequate for complex information-searching tasks. Navigation is not an efficient means of locating specific information, because users must painstakingly guide themselves using browsing cues: textual and graphical indications of the content reachable via a hyperlink. Searching, on the other hand, often returns inappropriate results and loses the important context present in the pages leading to the search result. By integrating navigation with searching and facilitating transition between them, users can locate desired information matching their complex information searching tasks effectively and efficiently.

- **Ranking Search Results for User Search**

Current algorithms for ranking Web pages, such as PageRank [Page et al. 1998; Brin and Page 1998] and HITS [Kleinberg 1998] consider only information about hyperlinks. They do not however take into account information about how users have used the hyperlinks. Therefore, their rankings of pages cannot fully reflect how users have used these hyperlinks. By taking into account link traversals as users' implicit feedback in using these links, we can provide rankings of Web pages that reflect users' views.

The third issue that this thesis addresses is to rank Web pages given past users' behavior in using the Web site link structure in order to help users search for desired information. A *collaborative approach* has been taken, which assumes that a user behaves in a similar way to other users. Rankings of pages can be obtained using data from a group of users, and they are then used to rank search results for an individual user in the absence of information about the user. When a user visits a Web site, rankings reflecting the collective behavior of the user group are used to rank search results. Both hyperlinks and user traversals on hyperlinks are taken into account in ranking Web pages respectively.

1.2 Overview of Our Approaches to Adaptive Web Site Navigation and Search

To address the above three issues, we have developed a number of approaches. First, we have developed approaches to mining information about Web site contents, hyperlinks, and user behavior contained in a Web site link structure to construct the link hierarchy and conceptual link hierarchy of a Web site. These hierarchies can then be visualized for adaptive user navigation. Second, we construct Markov chain models for link prediction from a Web site link structure, link hierarchy, and conceptual link hierarchy respectively. Third, we propose the PageRate algorithm to rank Web pages by taking into account Web site contents, hyperlinks, and user behavior contained in a Web site link structure or link hierarchy in assisting users' keyword based information searching on the Web site respectively. Searching on the link hierarchy is integrated with the visualized link hierarchy for adaptive user navigation and search.

1.2.1 Mining Web Site Link Structures for Link Hierarchies and Conceptual Link Hierarchies

Web sites with a hierarchical organization of Web pages are convenient for Web site designers to organize information and efficient for users to navigate among the Web pages that have been divided onto different conceptual levels. Users enter a Web site, go through multiple conceptual levels of the Web site and find desired information in one or multiple Web pages. Usually, users decide which hyperlink to follow on each Web page based on the anchor text or extended anchor text of the hyperlinks. The contextual information, e.g., the relationships between the current page and other Web pages on the Web site, which could help users navigate, is not visible to users. By visualizing the Web site in a hierarchy, users can get better informed about the contextual information of the current page and then decide where to go next. On the other hand, users are often limited to following the hyperlinks provided by the designer and using the navigation functions provided by Web browsers, e.g., “Back” and “Forward” button. As Nielsen [2000] noted, Web browsers are not sufficient for navigating the Web. By using the visualized hierarchy of the Web site, users can control their own navigation and take paths that have not been intended by the designer.

Munzner [2000], Durand and Kahn [1998], and many others have visualized a Web site in a hierarchy for Web site navigation. Both Munzner [2000] and Durand and Kahn [1998] constructed the hierarchy of a Web site based on the Web site designer’s views of the Web site. While in our approaches, the Web site link structure is adapted to users’ behavior, i.e., the constructed hierarchy is based on the users’ views of the Web site.

User traversals on hyperlinks can be seen as users’ collective views of the hyperlinks. The more user traversals on a hyperlink between two pages, the more closely the two pages are conceptually related to each other. We can construct the hierarchy of a Web site based on user traversals on hyperlinks. The hierarchy can reflect users’ views of the Web site. When users’ views change over time in terms of changes on user traversals, the constructed hierarchy can reflect these changes accordingly.

The visualized hierarchy helps users on their visits to the Web site. However, for a Web site consisting of a large number of pages and links, the visualized hierarchy can still be too large for users to manage and use. In order to tackle this large volume of visual information, we have developed two approaches. First, we try to reduce page clutter on the link hierarchy. We cluster conceptually related Web pages on each conceptual level of the link hierarchy thus the number of nodes in the link hierarchy can be dramatically reduced. Second, we try to synthesize titles for pages and clusters to give users an overview of the contents of the pages and clusters. We synthesize titles for pages and clusters using extended anchor texts of pages.

The study of hyperlinks between Web pages is closely related to the bibliographic analysis of citation links between scientific publications. Bibliographic analysis based on the notions of co-citation and coupling has been used to study the conceptual relationships between publications citing each other. It has now been extended to the study of conceptual relationships between Web pages linked with each other [Almind and Ingwersen 1997; Giles et al. 1998; Davison 2000; Henzinger 2000 and 2001]. We propose the notion of *link similarity*, including in-link and out-link similarities between Web pages on the same conceptual level of the link hierarchy. We develop a hierarchical clustering algorithm called *PageCluster* based on in-link and out-link similarities to cluster Web pages on each conceptual level of the link hierarchy to form two kinds of conceptual clusters, namely, *navigation* and *category clusters*. *Virtual links* are created between clusters and pages based on the conceptual relationships between them in a conceptual link hierarchy.

Extended anchor texts of hyperlinks to a Web page can be seen as objective conceptual descriptions of the Web page [Glover et al. 2002a and 2002b]. We use a feature vector to represent each extended anchor text and synthesize them together to get a feature vector to describe the Web page. For each cluster, feature vectors of its member pages are synthesized to create a feature vector for describing the cluster.

1.2.2 Markov Chain Models for Link Prediction

The navigation process of a user on a Web site can be modeled as a *Markov chain*, i.e., the pages that the user is likely to request in the future are determined by the pages already requested by the user. We assume that the collective navigation behavior of a group of users can be used to predict the navigation behavior of an individual user. Under this collaborative assumption, we can construct a Markov chain model of user navigation behavior using user traversals on hyperlinks as their collective behavior. The Markov chain model is then used to predict pages that an individual user is likely to request in the future based on pages already requested by the user.

We construct three Markov chain models from the Web site link structure, link hierarchy, and conceptual link hierarchy, respectively. These Markov chain models are then used for link prediction. Link prediction on link hierarchy and conceptual link hierarchy is further integrated with the visualized link hierarchy and conceptual link hierarchy. The three Markov chain models are evaluated and compared in terms of their effectiveness and efficiency in assisting user navigation.

- **Constructing Markov Chain Models from Web Site Link Structures (MMSs)**

In constructing a Markov chain model from a Web site link structure, we view the Web pages in the Web site link structure as states and the hyperlinks between Web pages as one-step transitions between them. User traversals on hyperlinks recorded in Web log files are used to estimate one-step transition probabilities between these states to form a transition matrix.

To predict the nth step in the future, we need to compute the$n$th power of the transition matrix, which can be computationally expensive given the large number of pages on a Web site. Thus we employ a transition matrix compression algorithm [Spears 1998] to reduce the size of the state space of the Markov chain model while retaining the accuracy of link prediction. States with the most similar transition behaviors measured by a similarity measure are aggregated into one state [Spears 1998]. The compressed transition matrix does not result in a significant increase of errors when being raised to a higher power.

- **Constructing Markov Chain Models from Link Hierarchies (MMHs)**

In a link hierarchy, secondary links between pages are removed since they have only auxiliary navigation functions. On the other hand, structural links between pages are kept since they have primary navigation functions and represent conceptual relationships between pages. We can view a Markov chain model constructed from the link hierarchy as a refined version of the Markov chain model constructed from the Web site link structure, in which transitions representing secondary links have been removed. By doing so, we can reduce the negative influence of secondary links in link prediction and focus on the effects of structural links.

- **Constructing Markov Chain Models from Conceptual Link Hierarchies (MMCs)**

A Markov chain model constructed from the conceptual link hierarchy can be seen as a compressed version of the Markov chain model constructed from the link hierarchy, in which states representing conceptually-related pages have been aggregated into one state representing a conceptual cluster. New transitions between aggregated states and other states are created in the Markov chain model representing virtual links between clusters and pages in the conceptual link hierarchy.

- **A Method for Using MMHs and MMCs for Efficient Link Prediction**

We present a method for computing the n th power of the transition matrix of a MMH or MMC. Instead of raising the transition matrix to its n th power, we multiply a series of transition matrices representing transitions between every two adjacent conceptual levels of a link hierarchy or conceptual link hierarchy. The complexity $O(N^3)$ of computing the n th power of the transition matrix can be reduced to the complexity $O(N'^3)$, where N is the number of states in the transition matrix and N' is the largest number of states on any conceptual level. Thus the n th power of the transition matrix can be computed more efficiently.

- **Link Prediction Using Markov Chain Models**

Sarukkai [2000] proposed a method to predict the most probably to-be-visited Web page in the next step by a user given a sequence of visited pages by the user. Based on his work, we propose a method to predict the most probably to-be-visited (MPT) Web page within the next n steps by a user given a sequence of visited pages by the user. Our link prediction can provide more insights into the future by taking into account more steps in the future. Users are not limited to one of the pages linked by the current page. They can directly jump to a page that can be many links away from the current page.

Chen et al. [1998] proposed a *maximal forward path method* to remove pages in a sequence of visited pages by a user, which are visited mainly for ease of navigation. The method removes backward links in a sequence of pages. We use this method to process a sequence of visited pages to get a maximal forward sequence that is then used in link prediction. This improves accuracy of link prediction.

Due to the hierarchical nature of the link hierarchy and conceptual link hierarchy, link prediction can be enhanced in three aspects.

First, the maximal forward path method [Chen et al. 1998] can be improved and used to process a sequence of visited Web pages and clusters to get a maximal forward sequence.

Second, we propose a method to predict *guided paths* for a user given his/her current Web page on a link hierarchy or conceptual link hierarchy. A guided path consists of a series of Web pages and clusters on consecutively adjacent conceptual levels of a link hierarchy or conceptual link hierarchy.

Third, link prediction results are integrated with the visualized link hierarchy or conceptual link hierarchy.

- Paths from the current page to the predicted page are shown and highlighted on a hierarchy.

- Predicted pages are highlighted on a hierarchy in proportion to their probabilities.

- Links pointing to predicted pages on the current page are highlighted in proportion to their probabilities.

- Guided paths starting at the current page are highlighted on a hierarchy.

1.2.3 Ranking Search Results for Adaptive Web Site Search

Silverstein et al. [1999] observed that users generally only view a few dozens of search results. Bharat and Broder [1998] observed a similar phenomenon in the AltaVista search query log. Thus, rankings of search results are crucial for users to find desired information effectively and efficiently.

Given a user's keyword-based query, traditional relevance-based information retrieval methods [Hand et al. 2001] only consider how relevant the content of a Web page is to a query. Typically, the content of a Web page is represented as a feature vector containing the weights of a set of features. The query is also represented as a feature vector. The *relevance-based ranking* of the Web page is defined as a similarity measure between the two feature vectors. The Web pages are ranked according to their relevance-based rankings in the search results.

Relevance-based rankings are not sufficient for Web searching. Web queries are generally very short (two to three terms). They can match hundreds and thousands of pages on the Web. Relevance-based rankings are also susceptible to problems, e.g., keyword spammings that are used to artificially promote the rankings of some pages [Chakrabarti 2002].

In bibliographic analysis, the number of citations to a paper is an indicator of its authority or prestige. Page et al. [1998] applied a variant of this idea and proposed the *PageRank* algorithm to rank Web pages using information about hyperlinks between pages. In the PageRank algorithm, a hyperlink from page A to page B is seen as a recommendation of page B by the author of page A. If page A is an authority, the recommendation of page B is authoritative. PageRank of a page is given by the PageRanks of those pages that are linked to it. The PageRanks of these pages are again given by the PageRanks of pages that are linked to them. Hence, PageRank of a page is determined recursively by the PageRanks of other pages.

The PageRank of a page may also be considered as the probability that a random surfer visits the page [Brin and Page 1998]. A surfer is given a page at random and clicks hyperlinks on the page and the following pages. At each page, the surfer may also get bored and jump randomly to a page. The probability for the surfer to keep clicking hyperlinks on a page is given by the damping factor d, which is set between 0 and 1. With probability d, the surfer decides to choose, uniformly at random, a hyperlink on each page. With probability ($1-d$), the surfer gets bored and jumps randomly to a page. The random surfing is equivalent to a random walk on the Web link structure. It can be shown that the PageRank of a page is proportional to the stationary distribution of the random walk on the Web link structure. The random surfer's navigation on the Web link structure can be modeled as a Markov chain. Ding et al. [2002] and Ng et al. [2001] suggested that the PageRanks of Web pages in a Web link structure can be obtained from the stationary distribution of a Markov chain model. The PageRank algorithm is used as a component of the Google search engine [Brin and Page 1998] to help determine how to order the pages returned by a Web search query.

Kleinberg [1998] proposed a similar algorithm called Hyperlink Induced Topic Search (HITS) to assign two scores to each page in a Web link structure. One is a measure of authority similar to the PageRank of a page, the other is a measure of a page being a hub, i.e., a comprehensive catalog of links to good authorities.

In addition to information about hyperlinks in a Web site link structure, we propose to also take into account user behavior in ranking Web pages so that the rankings of pages can be adapted to user behavior. Thus, as opposed to treating each link equally, in the PageRank and HITS algorithms, in distributing the ranking of a page to the pages that it is linked to, user behavior in following these links is taken into account in distributing the ranking of the page. We propose the PageRate algorithm as an improvement of the PageRank algorithm to rank Web pages, which uses information about both hyperlinks and user behavior in using a Web site link structure.

Link traversals recorded in Web log files can be seen as users' implicit feedback in using these hyperlinks. Each hyperlink on a page does not necessarily have the same level of importance. It is obvious that a hyperlink in a prominent position, in large font and in bold typeface is viewed as more important and will normally receive much more clicks from users than another hyperlink in an unnoticeable corner, in small font and in normal typeface. Web log files loyally record user behavior in using these hyperlinks.

We propose the PageRate algorithm to take into account users' implicit feedback, in the form of user traversals on hyperlinks, in ranking Web pages. In the PageRate algorithm, user traversals on hyperlinks affect PageRates of the pages in the Web site link structure. PageRate is biased toward the pages linked by frequently traversed hyperlinks. As opposed to treating each hyperlink in a page equally in the PageRank algorithm, user traversals are used to weight the hyperlinks in a page. The more users have traversed a hyperlink in a page, the more PageRate of the page is distributed to the page the hyperlink points to.

The PageRate algorithm is applied to the link structure and link hierarchy of a Web site to get PageRates of pages respectively. Extended anchor texts of hyperlinks to a Web page can be seen as more accurate and objective conceptual descriptions of the Web page than the contents of the Web page itself [Glover et al. 2002a and 2002b]. We use a feature vector to represent each extended anchor text and synthesize them together to get a feature vector to describe the Web page. A *relevance-based ranking* is defined between two feature vectors representing a page and the user query respectively. PageRates of pages are combined with their relevance-based rankings as their overall rankings, which determine how the pages returned by a user query are ordered.

Search on the link hierarchy is further enhanced by integrating search results with the visualized link hierarchy and link prediction on the link hierarchy. Two kinds of methods are used for integrating navigation with search.

First, contexts of search results within a link hierarchy are shown.

- Paths from the home page or current page to the search results are shown and highlighted on a link hierarchy.
- Search results are highlighted on a link hierarchy in proportion to their overall rankings.

Second, navigation on a link hierarchy can be used to facilitate search.

- Users can go through pages level by level and search for pages containing desired information to speed up the information searching process. Navigation is used when it is hard to describe the desired information with a few keywords. Alternatively, users can use search to find a page, which is used as the starting point for navigation.
- Users can reformulate their queries while navigating a link hierarchy.
- Links pointing to search results on the current page are highlighted in proportion to their overall rankings.
- Users can search part of a link hierarchy, e.g., a sub-hierarchy rooted at a user-specified page in the link hierarchy.

The work is a step toward integrating search with navigation in an adaptive Web site. Search on the Web site link structure and that on the link hierarchy are compared in terms of their effectiveness and efficiency in helping users find desired information on the Web site.

1.3 Overview of Experimental Evaluation

The University of Ulster Web site has been used in the evaluation, which consists of 3,546 pages and 3,953 links. The adapted Web site is presented in a prototype system called *Online Navigation Explorer*(ONE) to help users navigate the Web site and search for desired information. The results of our experiments are threefold.

First, we evaluated our approaches to visualizing Web site link structures. Our experiments show that our method can put pages onto the conceptual levels of a link hierarchy more accurately than both the breadth-first search method and the shortest weighted path method in link hierarchy construction. The PageCluster algorithm can cluster conceptually related pages more accurately than the bibliographic analysis method. The constructed conceptual link hierarchy, where pages on each conceptual level of the link hierarchy have been clustered, is more compact than the link hierarchy.

The conceptual link hierarchy is visualized in ONE for user navigation. ONE helps users control navigation by themselves, i.e., they are not confined to just following hyperlinks in each Web page. They can understand their current locations in the context of different conceptual levels consisting of Web pages and clusters in the conceptual link hierarchy. They can move up and down in the conceptual link hierarchy or jump from one Web page or cluster to another Web page or cluster. Our experimental results show that the conceptual link hierarchy visualized in ONE has given users a clearer view of their current locations on the Web site and the conceptual relationships among the Web pages and clusters. The conceptual link hierarchy visualized in ONE can help users find information more effectively and efficiently as the task of finding information becomes less specific and involves more Web pages on multiple conceptual levels.

Second, we evaluated our approaches to link prediction. Three Markov chain models are constructed from a Web site link structure (MMS), link hierarchy (MMH), and conceptual link hierarchy (MMC) for link prediction respectively.

- Most probably to-be-visited (MPT) pages within the next n steps, given a sequence of visited pages, are predicted using MMSs and presented in ONE for user navigation.

- MPT pages within the next n steps, given a sequence of visited pages and guided paths, given the current page, are predicted using MMHs and MMCs respectively. Link prediction results are integrated with the visualized link hierarchy and conceptual link hierarchy respectively for user navigation.

Our user study shows that link prediction using MMHs integrated with visualized link hierarchies helped users find desired information more effectively and efficiently than link prediction using MMSs. Link prediction using MMCs integrated with visualized conceptual link hierarchies helped users find desired information slightly more effectively and efficiently than link prediction using MMHs integrated with visualized link hierarchies.

Third, we evaluated our approaches to ranking search results. We compared the PageRate algorithm with the PageRank algorithm on searching the Web site link structure.

Search results ranked by the PageRate algorithm and the PageRank algorithm are presented in ONE to help users search for desired information respectively. Search results ranked by the PageRate algorithm are integrated with the visualized link hierarchy in ONE to help users navigate and search for desired information.

Our user study shows that search results ranked by the PageRate algorithm can help users search for desired information more effectively and efficiently than those by the PageRank algorithm. Search results ranked by the PageRate algorithm integrated with the link hierarchy can help users search for desired information more effectively and efficiently than search results ranked by the PageRate algorithm alone.

1.4 Summary of Contributions

This thesis presents approaches to mining Web site link structures for knowledge about Web sites and users, which is used to design adaptive Web sites to help users navigate a Web site and search for desired information effectively and efficiently. Our major contributions are fivefold.

First, we propose a novel method of constructing a link hierarchy of a Web site, which is adapted to user behavior. User traversals on hyperlinks between Web pages reveal conceptual relationships between these pages. On the basis of these traversals, our method puts Web pages on a Web site onto different conceptual levels in a link hierarchy. Our experiments show that our method can put Web pages onto conceptual levels of a link hierarchy more accurately than both the breadth-first search method and the shortest weighted path method.

Second, we propose the PageCluster algorithm to cluster conceptually related pages on each conceptual level of a link hierarchy. Page clutter is thus reduced in the link hierarchy. We define link similarity between Web pages on the same conceptual level as an improvement of bibliographic co-citation and coupling measures. Our experiments show that the PageCluster algorithm can cluster conceptually related pages more accurately than the bibliographic analysis method. Clusters and unclustered pages are used to construct a conceptual link hierarchy, which is visualized for adaptive Web site navigation. Our user study shows that the conceptual link hierarchy can help users find information effectively and efficiently.

Third, we propose to construct Markov chain models from Web site link structures (MMSs), link hierarchies (MMHs), and conceptual link hierarchies (MMCs) respectively for link prediction. We apply a compression algorithm proposed by Spears [1998] to compress the transition matrix of a MMS for efficient link prediction while retaining the accuracy of link prediction.

Fourth, we propose to improve link prediction using MMSs, MMHs, and MMCs, respectively, in three aspects.

- We apply the maximal forward path method [Chen et al. 1998] to a sequence of visited pages to get a maximal forward sequence of pages for accurate link prediction using MMSs. We improve the maximal forward path method and apply it to a sequence of visited pages and clusters to get a maximal forward sequence of pages and clusters for accurate link prediction using MMHs and MMCs, respectively.

- We propose to predict the most probably to-be-visited (MPT) Web pages and clusters within the next n steps given a sequence of visited pages and clusters. This is an improvement of Sarukkai's method for predicting MPT pages in the next step given a sequence of visited pages [Sarukkai 2000].

- We propose to predict guided paths given a user's current page or cluster using MMHs and MMCs, respectively. A *guided path* of length m consists of m linked Web pages or clusters on m adjacent conceptual levels on the link hierarchy or conceptual link hierarchy starting from the adjacent lower conceptual level of the current page or cluster, respectively.

Link prediction using MMHs and MMCs is integrated with visualized link hierarchies and conceptual link hierarchies for user navigation, respectively. Our experiments show that link prediction using all three Markov chain models helped users find desired information effectively and efficiently. Link prediction using MMHs integrated with visualized link hierarchies helped users find information more effectively and efficiently than MMSs. Link prediction using MMCs integrated with visualized conceptual link hierarchies helped users find information slightly more effectively and efficiently than MMHs integrated with visualized link hierarchies.

Fifth, we propose to adapt rankings of Web pages to user behavior. We propose the PageRate algorithm to take into account information about both hyperlinks and user behavior in ranking Web pages as an improvement of the PageRank algorithm [Page et al. 1998; Brin and Page 1998]. The PageRate algorithm is applied to the Web site link structures and link hierarchies, respectively. PageRates of pages are integrated with their relevance-based rankings as overall rankings of these pages used to determine how to order search results. Search results ranked using the PageRate algorithm are integrated with visualized link hierarchies for adaptive Web site navigation and search. Our experiments show that search results ranked by the PageRate algorithm helped users search for desired information more effectively and efficiently than those by the PageRank algorithm. Search results ranked using the PageRate algorithm integrated with link hierarchies helped users find desired information more effectively and efficiently than search results ranked using the PageRate algorithm alone.

1.5 Overview of the Rest of The Thesis

The rest of this thesis is organized as follows. In chapter 2, we discuss the background of the thesis. In chapter 3, we present approaches to mining link hierarchies and conceptual link hierarchies from Web log files for adaptive Web site navigation. In chapter 4, we present approaches to link prediction using Markov chain models for adaptive Web site navigation. In chapter 5, we present approaches to adapting search on a Web site to user behavior. In chapter 6, we evaluate our approaches presented in chapter 3, 4, and 5, respectively. Finally, in chapter 7, we conclude the thesis.

Chapter 2

BACKGROUND

Three types of information are available for mining on the Web, namely, information about hyperlinks, Web page contents, and Web site usage. In this chapter, we discuss related research on mining the Web for assisting user navigation and search. Our approaches are in the context of*Web mining*, which applies machine learning and data mining techniques to the Web for useful knowledge about the Web and its users.

Three major approaches have been proposed to tackle the navigation and search problems. First, information about hyperlinks has been used for assisting user navigation and search. The link structures of the Web and Web sites are visualized. Hyperlinks are used to cluster Web pages. Hyperlinks are used to get authoritative rankings of Web pages.

Second, information about Web page contents has been used for assisting user navigation and search. Web page contents are used to cluster Web pages. Anchor texts and extended anchor texts have been shown more useful than Web page contents in classification of Web pages. Anchor texts and extended anchor texts have also been used to extract the titles of pages and clusters. Web page contents, anchor texts, and extended anchor texts have been used to get relevance-based rankings of Web pages.

Third, information about Web site usage has been used for assisting user navigation. Web site usage data, which contain records of how users have visited a Web site, have been used to identify collective user behavior in using the Web site. Markov chains have been widely used to model user navigation on the Web. Markov chain models are constructed using the information about user behavior in traversing links and then used to predict user navigation on the Web site. Adaptive Web sites are proposed which can change the organization and presentation to help users navigate and search for desired information on the Web sites.

2.1 *Utilizing Information about Hyperlinks*

The hyper-linking nature of the Web has differentiated itself from traditional forms of information resources. Hyperlinks are created between Web pages by Web page authors mainly to assist users in navigating the vast amount of information on the Web. Hyperlinks can reveal conceptual relationships between linked Web pages. Mining these relationships in the link structure of a Web site can help users navigate and search for desired information on the Web site effectively and efficiently. First, the link structure of a Web site is visualized for user navigation. Second, Web pages are clustered to reduce information overload problem in user navigation. Third, hyperlinks are used to get authority-based rankings of Web pages for user information search on the Web.

2.1.1 Visualizing Web Site Link Structures for Navigation

The link structure of a Web site can be visualized in a *link hierarchy* consisting of Web pages on multiple conceptual levels for user navigation. The advantage of the link hierarchy is that it gives users a clear view of their locations on the Web site relative to the other Web pages and conceptual levels in the hierarchy.The breadth-first search method is commonly used to construct a link hierarchy of a Web site. In the breadth-first search method, starting at the home page, links are traversed level by level. All links from each of the Web pages on the current conceptual level are traversed, leading to the Web pages on the next conceptual level. In other words, each page is put onto a conceptual level of the link hierarchy determined by the shortest path from the home page to the page. The distance from the home page to a page is measured by the number of links between them. The problem with this method is that the shortest path from the home page to a page does notnecessarily match the intended navigation path from the home page to the page.

Durand and Kahn [1998] developed a system called MAPA to extract a hierarchical structure from an arbitrary Web site for navigation. They argued that hierarchical structures restrict the form of the underlying graph by excluding hyperlinks across non-adjacent conceptual levels and from a lower conceptual level to a higher conceptual level. In MAPA, a heuristically-determined and user-assigned weight is associated with each hyperlink between two pages to reflect the conceptual relatedness between the two pages. The lower the weight on a link between two pages is, the more the two pages are conceptually related. Each page is put onto a conceptual level of the link hierarchy determined by the "minimum-weight" path from the home page to the page in the Web site. Dijkstra's algorithm [Cormen et al. 1990] for the single-source shortest path problem is used in MAPA to find the "minimum-weight" paths from the home page to every other page in the Web site.

Munzner [2000] proposed the *H3* (Hyperbolic 3D) layout system to find a spanning tree from the link structure of a hierarchical Web site. She proposed that for a Web site designer most pages have conceptually a main parent even though there might be multiple pages linked to it. The hierarchical directory structure of a Web site encoded in the URLs of pages is used to determine which of the pages linked to a page should be chosen as its main parent. Each page is put onto a conceptual level of the link hierarchy determined by its main parent. However, the assumption that the hierarchical directory structure of a web site reflects the hierarchical organization of the Web site is not always true, since the site designer may have not created the directory structure to reflect the organization of a Web site.

In chapter 3, we propose a method to construct the link hierarchy of a Web site on the basis of user usage data contained in a Web log file. In our method, as opposed to assigning weights to links manually in MAPA, we treat the number of user traversals on a link as collective user feedback in using the link and assign it to the link as the weight on the link. The higher the weight on a link between two pages is, the more the two pages are conceptually related. Among all links to a page, the link with the highest weight is chosen as the *main link* to the page and the page where the main link comes from is the *main parent* of the page accordingly. Similar to the method proposed by Munzner [2000], we put a page onto a conceptual level of the link hierarchy determined by its main parent. A page may have multiple pages as candidates for its main parent if these pages have an equal highest weight on their links to the page. Whichever of these candidates appears first is chosen as the main parent of the page.

2.1.2 Clustering Web Pages for Navigation

However, for a Web site that has a considerable number of Web pages and hyperlinks, the link hierarchy can still be too large and complex for users to navigate. Some work has been done to reduce page clutter by clustering Web pages using hyperlinks. Kleinberg [1998] proposed the *HITS* algorithm to find *hubs* and *authoritative pages* based on an adjacency matrix derived from links between Web pages. Cyber-communities sharing common interests are identified through link analysis [Kumar et al. 1999]. Flake et al. [2002] defined a *Web community* as a collection of Web pages in which each member page has more hyperlinks within the community than outside the community. They argued that the creation of a hyperlink is a stronger indication of relevance between Web pages than content-based similarity.

2.1.2.1 Link based Similarity of Web Pages

The study of hyperlinks between Web pages has a lot of similarity to the bibliographic analysis of citation links between scientific publications. In bibliographic analysis, *coupling* of two publications is measured by the number of publications they have cited in common [Kessler 1963]. *Co-citation* of two publications is measured by the number of times they have been cited together in other publications [Small 1973]. Co-citation and coupling can be used as two basic similarity measures for clustering. Carpenter and Narin [1973] clustered scientific journals into sub-disciplines based on co-citation. Small and Koenig [1977] clustered scientific journals based on coupling. Kessler [1963] showed that a clustering based on coupling yields meaningful groupings of publications.

Almind and Ingwersen [1997] contended that WWW is a citation network. Hyperlinks between Web pages are similar to citation links between publications, in that one Web page refers to another in the similar way that one publication cites another. Pitkow and Pirolli [1997] observed that hyperlinks, when employed in a non-random format, provide semantic linkages between Web pages, in much the same manner that citations link publications to other related publications. Co-citation has been used as a measure of the similarity between Web pages [Larson 1996; Pitkow and Pirolli 1997].

Dean and Henzinger [1999] developed an algorithm for finding Web pages that have co-citations with a given page. These co-citations occur close to each other in the pages citing them.

In chapter 3, we define in-link and out-link similarities instead of co-citation and coupling similarities. Co-citation and coupling similarities of two Web pages are defined as the number of common in-links and out-links that the two pages have respectively, where each in-link and out-link is treated equally in measuring similarities. We use the numbers of user traversals on the links as collective user feedback in using these links. User traversals on the in-links and out-links of a Web page are used to represent the in-link and out-link strengths of the page respectively. We can interpret link strength as follows. Web masters generally put those links to Web pages, which they think the most relevant to the current page, in the most prominent positions in the current page. These links attract most users due to their prominence. The more users have traversed these links, the higher the link strengths on these links are. We define the *in-link similarity* of two pages as a distance-based measure of the link strengths on their common and uncommon in-links. We define the *out-link similarity* of two pages as a distance-based measure of the link strengths on their common and uncommon out-links. Such defined link similarities can incorporate user preferences and thus are more objective and user-centric.

In-link and out-link similarities are more suitable for Web page clustering than co-citation and coupling similarities. First, since some Web pages have only two or three in-links and out-links, co-citation and coupling similarities are too coarse for clustering and there could be a very large number of pairs of pages having the same similarity measures. Second, in-link and out-link similarities reflect collective user behavior in a given time period. Two pages having link similarity in one period may not be so in another period. Different clusters can be generated to reflect user behavior changes. These changes cannot be reflected in the clusters generated on the basis of co-citation and coupling similarities.

2.1.3 Utilizing Information about Hyperlinks for Search

In traditional content-based information retrieval methods, a feature vector representing the contents of a document is compared with a feature vector representing a user query to decide the relevancy of the document to the query. A*content-based similarity measure* is defined between the two feature vectors as the relevance-based ranking of the document. The most relevant documents judged by their relevance-based rankings are returned as the search results of the query. In the search results, documents are ordered on the basis of their relevance-based rankings.

However, while searching the Web, traditional information retrieval methods cannot guarantee that the returned documents are authoritative on the query topic. Most Web queries consist of a few keywords. Searches typically return hundreds and thousands of documents, among which there are only a few authoritative ones.

On the other hand, due to the hyper-linking nature of the Web, link analysis can be used to study hyperlinks in order to improve the quality of search results. The way Web page authors use hyperlinks can give valuable information about Web pages. Anchor texts of a Web page can be viewed as more accurate descriptions of the page than the page content itself. World Wide Web Worm [Oliver A. McBryan 1994] associated anchor texts with the Web page the anchor texts point to. In a search, keywords are compared with a collection of anchor texts that point to a Web page, rather than a feature vector of the Web page content. A rank is then assigned to the page based on the degree to which the keywords match the anchor texts of the page.

Furthermore, hyperlinks alone can indicate how authoritative a Web page is. A link from page A to page B can be seen as a recommendation of page$B$ by the author of page A. Generally, the number of Web pages linked to a Web page can be used to measure the quality of the page. In bibliographic analysis, citation counting is a simple method for determining the importance of a publication by counting the number of citations to it. In the case of scientific publications that have relative uniform quality and importance, we can assume that a highly cited publication should be of greater interest than a publication with only a few citations. In the case of Web pages whose contents vary greatly in quality and importance, citation counting is over-simplistic.

Page et al. [1998] proposed the PageRank algorithm to rank Web pages by not only the number of citations they have, but also the importance of pages that cite them. In the PageRank algorithm, a Web page linked by a highly important page only, e.g., the home page of Yahoo, can be more important than another page linked by many unimportant pages, e.g., many personal home pages. PageRanks of pages can be integrated with their relevance-based rankings as their overall rankings to rank search results.

PageRanks of pages can be understood as a steady-state probability distribution calculated from a model of a random surfer on the Web link structure [Brin and Page 1998]. A surfer is given a page at random and clicks each hyperlink on the page with equal opportunity decided by the number of hyperlinks on the page. The surfer keeps clicking links on following pages. With probability d, the damping factor, the surfer decides to choose, uniformly at random, a hyperlink in each page. With probability (1-d), the surfer jumps to a random page on the Web link structure. The random surfing is equivalent to a random walk on the Web link structure. The random walk problem is a well-studied combinatorial problem [Motwani and Raghavan 1995]. It can be shown that the PageRank of a page is proportional to the stationary distribution of the random walk on the Web link structure. Thus, the PageRank of the page is proportional to the frequency with which a random surfer will visit the page. The probability that the random surfer visits a page A is its PageRank.

Alternatively, the PageRank of a page is determined recursively by the PageRanks of other pages. According to Page et al. [1998], PageRank of a page A is:

$$PR(A) = \frac{1-d}{N} + d \cdot \left\{ \frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)} \right\} \tag{2.1}$$

where $PR(A)$ is the PageRank of page A, N is the number of pages in the Web link structure, $PR(T_i)$ is the PageRank of page T_i which links to page A, $C(T_i)$ is the number of links in page T_i, and d is the damping factor which can be set between 0 and 1. The PageRank of page A is recursively defined by the PageRanks of those pages that are linked to page A.

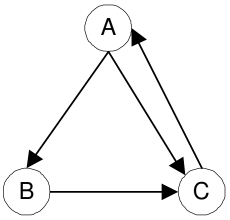


Figure 2.1: A link structure for PageRank calculation.

In Figure 2.1, a link structure consisting of three pages A, B, and C, where page A is linked to pages B and C, page B is linked to page C, and page C is linked to page A. Suppose the damping factor d is 0.5. We get the following linear equations for the PageRank calculation:

$$\begin{cases} PR(A) = 0.5/3 + 0.5 \cdot PR(C) \\ PR(B) = 0.5/3 + 0.5 \cdot (PR(A)/2) \\ PR(C) = 0.5/3 + 0.5 \cdot (PR(A)/2 + PR(B)) \end{cases}$$

We can solve these equations to get the PageRanks of the pages as follows:

$$\begin{cases} PR(A) = 14/39 \approx 0.36 \\ PR(B) = 10/39 \approx 0.26 \\ PR(C) = 15/39 \approx 0.38 \end{cases}$$

However, for a Web link structure consisting of a very large number of pages, it is not possible to find a solution to a large set of linear equations by inspection only. Accordingly, an iterative procedure is used. At the initial state we may simply set the PageRanks of all pages equal to $1/N$. The linear equations are then used to calculate a new set of PageRanks based on the current PageRanks. For the link structure in Figure 2.1, the iteration process is shown in Table 2.1.

Table 2.1: Iterations in PageRank calculation.

Iteration	$PR(A)$	$PR(B)$	$PR(C)$
0	0.333	0.333	0.333
1	0.334	0.250	0.417
2	0.376	0.251	0.373
3	0.354	0.261	0.385
4	0.360	0.256	0.384
5	0.359	0.257	0.384
6	0.359	0.257	0.384

We can see that we get a good approximation of the PageRanks after only a few iterations. Brin and Page [1998] suggested that in the case of millions of pages, sufficient convergence typically takes on the order of 100 iterations.

The random surfer's navigation on the Web link structure can be seen as a Markov chain. In the Markov chain model, the probability of following each hyperlink in a page is the one-step transition probabilities from a state representing a page to the states representing the pages that the page is linked to in a transition matrix Q. Writing down Equation 2.1 for all N Web pages, we get the vector v of the PageRanks of pages:

$$v = d \cdot (Q^T \times v) + (1-d) \cdot e \qquad (2.2)$$

where e is a $1 \times N$ vector such that $e = (1/N, \ldots, 1/N)$.

This can also be written in the form of a new transition matrix Q' that defines v:

$$v = (d \cdot Q^T + (1-d) \cdot M) \cdot v = (Q')^T \cdot v \qquad (2.3)$$

where M is a $N \times N$ matrix and $\forall i, j$, $M_{i,j} = 1/N$.

Now v is the eigenvector of the transition matrix Q', and v is also the stationary distribution of a Markov chain model, which has Q' as its transition matrix.

However, the PageRank algorithm still leaves space for improvements. As opposed to treat each hyperlink in a page equally in the PageRank algorithm, Pitkow et al. [2002]

proposed to incorporate link usage data in ranking Web pages. They argued that by computing what the most respected authors deem important, citation and link analysis approaches provide an implicit measure of importance of Web pages. However, these techniques can create an authoring bias where the meaning and resources valued by a group of authors determine the results for the entire user population. A ranking bias can occur when, for a given topic, the authoring community values a set of resources different from the general user population. A typical example of this is link promotion where a set of highly interconnected Web sites is created by a small set of authors in an attempt to appear to become the most relevant resources on a particular topic. Usage based ranking methods improve the previous link based ranking methods by leveraging the opinions of users to compute relevancy. A usage ranking provides a direct measure of what is relevant at any point in time to the users. The usage ranking of a page can be combined with content and link-based rankings of the page. Pitkow et al. presented the Outride system, which incorporates user usage data in ranking pages. Outride can assist users to perform information-searching tasks more effectively than current search engines, such as Google that uses PageRank as its fundamental ranking algorithm.

In chapter 5, we propose the PageRate algorithm as an improvement of the PageRank algorithm to take into account a group of users' implicit feedback in using links. As opposed to treat each hyperlink in a page equally in the PageRank algorithm, user link traversals are used to weight the hyperlinks in a page. The more users have traversed a hyperlink in a page, the more PageRate of the page is distributed to the page the hyperlink links to. The PageRate of a page A is:

$$PR(A) = \frac{1-d}{N} + d \cdot \left\{ PR(T_1) \cdot \frac{n(T_1, A)}{\sum_i n(T_1, i)} + \ldots + PR(T_n) \cdot \frac{n(T_n, A)}{\sum_i n(T_n, i)} \right\} \tag{2.4}$$

where $n(T_j, A)$ is the number of user traversals on the hyperlink from page T_j to page A. $\sum_i n(T_j, i)$ is the sum of user traversals on all the hyperlinks of page T_j.

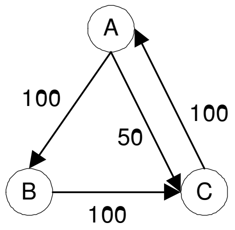


Figure 2.2: The link structure for PageRate calculation.

In Figure 2.2, a link structure consists of three pages$A$, B, and C, where page A is linked to pages B and C, page B is linked to page C, and page C is linked to page A. The numbers of user link traversals are shown beside the hyperlinks. We set damping factor d as 0.5, and get the following linear equations for the PageRate calculation:

$$\begin{cases} PR(A) = 0.5/3 + 0.5 \cdot \left\{ PR(C) \cdot \dfrac{100}{100} \right\} \\ PR(B) = 0.5/3 + 0.5 \cdot \left\{ PR(A) \cdot \dfrac{100}{100+50} \right\} \\ PR(C) = 0.5/3 + 0.5 \cdot \left\{ PR(A) \cdot \dfrac{50}{100+50} + PR(B) \cdot \dfrac{100}{100} \right\} \end{cases}$$

These equations can easily be solved. We get the following PageRates of the pages:

$$\begin{cases} PR(A) = 7/20 \approx 0.35 \\ PR(B) = 17/60 \approx 0.28 \\ PR(C) = 11/30 \approx 0.37 \end{cases}$$

We can see that the PageRates of the pages in Figure 2.2 are different from the PageRanks of the pages in Figure 2.1. PageRates of pages can reflect user behavior changes in terms of their traversals on hyperlinks, while PageRanks cannot.

PageRates of pages can also be represented using Equation 2.2 and 2.3. Numbers of user link traversals are used for computing one-step transition probabilities between states in transition matrix Q. PageRates of pages are also the stationary distribution of a Markov chain model, which has $Q' = d \cdot Q^T + (1-d) \cdot M$ as its transition matrix.

2.2 Utilizing Information about Web Contents

Information about Web contents has been utilized to help users navigate and search the Web. First, Web contents consisting of contents of Web pages, anchor texts, and extended anchor texts have been used to cluster Web pages for navigating among a large number of documents on the Web. Second, Web contents are used to get relevance-based rankings of Web pages, which are combined with authority-based rankings of the pages to rank search results. Third, feature vectors are extracted from Web contents to represent Web pages. Web pages are described and classified using these feature vectors.

2.2.1 Clustering Web Pages

Web page contents have been used for clustering Web pages. Crouch et al. [1989] used a vector to represent the contents of a Web page as a set of weighted terms. The content similarity of two Web pages is defined as a distance-based measure between the two vectors representing the two pages respectively. A Web page is initially treated as a singleton cluster and used as an input to an agglomerative hierarchical clustering algorithm, in which clusters are combined sequentially based on the similarity measure between them. The agglomerative hierarchical clustering and k-means algorithms have been used, in the Scatter/Gather system developed by Cutting et al. [1992 and 1993], to cluster documents on the basis of their contents for browsing large information spaces. Scatter/Gather presents summaries of clusters to users, who can then select some of these clusters for re-clustering the documents in them. New clusters become smaller and their contents are revealed in more detail.In chapter 3, we propose an agglomerative hierarchical clustering algorithm called PageCluster, which clusters conceptually related pages based on their in-link and out-link similarities. The pages in the same cluster may not be similar in their contents, and therefore cannot be clustered together by the content-based methods.

2.2.1.1 Representing Web Pages and Content based Similarity of Web Pages

The content of each Web page can be seen as a list of features. A feature can be a word, word-pair, or phrase. We remove stop-words which are deemed irrelevant, e.g., “a”, “the”, “of”. We combine features with a common word stem together as a single feature, e.g., “mined”, “mine”, and “mining”. We get a set of unique features from a collection of pages on the Web. We set a threshold to remove those features, which do not occur frequently, from the set. The size of a feature set can be quite large. For a set of T features, t_j, $1 \le j \le T$, a Web page D_i, $1 \le i \le N$, in the collection of pages can be represented as a feature vector, $D_i = (d_{i1}, d_{i2}, ..., d_{iT})$, where d_{ij} is a feature weight associated with the occurrence of the jth feature in the ith document. In the Boolean representation [Hand et al. 2001], a feature weight simply indicates whether a feature occurs in the document, i.e., $d_{ij}=1$ if document i contains feature j, and $d_{ij}=0$ otherwise. In the vector space representation [Hand et al. 2001], a feature weight can be a real number, e.g., a function of how often the feature occurs in the document.

For example, we have 10 documents and 6 features, which are t_1=“data”, t_2=“mining”, t_3=“data mining”, t_4=“knowledge”, t_5=“discovery”, t_6=“knowledge discovery”. We have a 10×6 document-feature frequency matrix M, where entry i, j contains the number of times feature j occurs in document i, as shown in Table 2.2.

Table 2.2: Document-feature matrix for 10 documents and 6 features.

	t_1	t_2	t_3	t_4	t_5	t_6
d_1	24	21	9	0	0	0
d_2	32	10	5	0	3	0
d_3	12	16	5	0	0	0
d_4	6	7	2	0	0	0
d_5	43	31	20	0	3	0
d_6	2	0	0	18	7	16
d_7	0	0	1	32	12	0
d_8	3	0	0	24	4	2
d_9	1	0	0	34	27	25
d_{10}	6	0	0	17	4	23

Given the feature vector of document$i$, $D_i = (d_{i1}, d_{i2}, ..., d_{iT})$ and the feature vector of document j, $D_j = (d_{j1}, d_{j2}, ..., d_{jT})$, the content based similarity between two Web pages as the cosine distance [Hand et al. 2001] between the two feature vectors is:

$$Sim_c(D_i, D_j) = \frac{\sum_{k=1}^{T} d_{ik} \cdot d_{jk}}{\sqrt{\sum_{k=1}^{T} d_{ik}^2 \cdot \sum_{k=1}^{T} d_{jk}^2}} \qquad (2.5)$$

2.2.1.2 Hierarchical Clustering Algorithms

Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging (agglomerative hierarchical clustering as a bottom-up scheme) or splitting (divisive hierarchical clustering as a top-down scheme) clusters based on a similarity (or dissimilarity) measure [Jain and Dubes 1988]. Agglomerative hierarchical clustering algorithms start with individual objects. Thus, there are initially as many clusters as objects. The most similar objects are first grouped, and these initial groups are merged according to their similarities. Eventually, as the similarity decreases, the process continues until a stopping criterion is met or all subgroups are grouped into a single cluster. Divisive hierarchical clustering methods work in the opposite direction. An initial single group of cluster is divided into two subgroups such that the objects in one subgroup are “far from” the objects in the other. These subgroups are then further divided into dissimilar subgroups; the process continues until a stopping criterion is met or there are as many subgroups as objects, i.e., each object forms a group. The results of both agglomerative and divisive methods may be displayed in the form of a two-dimensional diagram known as a dendrogram. The dendrogram represents the nested groupings of objects and similarity levels at which groupings change. The dendrogram can be broken at different levels to yield different clusterings of the objects.

The single-linkage (minimum distance or nearest neighbor) [Sneath and Sokal 1973] and complete-linkage (maximum distance or farthest neighbor) [King 1967] are two popular hierarchical clustering methods. These two methods differ in the way they characterize the similarity between a pair of clusters. In the single-linkage method, the distance between two clusters is the minimum of the distances between all pairs of objects drawn from the two clusters (one object from each cluster). In the complete-linkage method, the distance between two clusters is the maximum of all pairwise distances between objects in the two clusters. In either method, two clusters are merged to form a larger cluster based on minimum distance criteria. The complete-linkage method produces tightly bound or compact clusters [Baeza-Yates 1992]. The single-linkage method produces chain-like clusters, and thus suffers from a chaining effect [Nagy 1968]. It has been observed that the complete-linkage method produces more useful hierarchies in many applications than the single-linkage method [Jain and Dubes 1988].

The general process of the agglomerative hierarchical clustering algorithm consists of three steps as follows.

- First, compute the similarity (or dissimilarity) matrix containing the similarity (or distance) between each pair of objects. Treat each object as a cluster.

- Second, find the most similar pair of clusters using the similarity (or dissimilarity) matrix. Merge these two clusters into one cluster. Update the similarity (or dissimilarity) matrix to reflect this merge operation.

- Third, if a stopping criterion is met or all objects are in one cluster, stop. Otherwise, go to step 2.

In Step 1, one common distance measure is *Minkowski metric* [Hand et al. 2001]:

$$d(X,Y) = [\sum_{i=1}^{d} | x_i - y_i |^m]^{1/m} \tag{2.6}$$

When $m = 1$, $d(X,Y)$ represents the city-block distance between two objects [Hand et al. 2001]. When $m = 2$, $d(X,Y)$ represents the Euclidean distance [Hand et al. 2001]. In general, varying m changes the weight given to larger and smaller differences.

Pairs of objects are often compared on the basis of the presence or absence of certain characteristics. Similar objects have more characteristics in common than dissimilar objects. The presence or absence of a feature can be described mathematically in a coefficient measure. Given two objects i and j both having a set of d features, f_k, $1 \le k \le d$, a general *similarity coefficient measure* [Gower 1971] S_{ij} between i and j is:

$$S_{ij} = \frac{\sum_{k=1}^{d} w_{i,j,k} \cdot S_{i,j,k}}{\sum_{k=1}^{d} w_{i,j,k}} \tag{2.7}$$

where $S_{i,j,k}$ represents the similarity between the two objects on the k th feature. $w_{i,j,k}$ is a user-defined weight for the k th feature but is set to zero when no valid comparison is possible between two objects on the feature. The value of $S_{i,j,k}$ can be defined for different types of features, e.g., in text mining, the similarity of two documents can be measured by the cosine of the angle between the two feature vectors representing the two documents. Alternatively, we can use a coefficient similarity measure, where $S_{i,j,k} = 1$ if both documents contain a certain feature and 0 otherwise. We can set $w_{i,j,k} = 1$ if at least one of them has the feature and 0 if none of them has the feature.

The complete-linkage and single-linkage methods use different ways to update the similarity (or dissimilarity) matrix in Step 2. The single-linkage method is more susceptible to outliers or noisy data points than the complete-linkage method. There are many different variations of agglomerative hierarchical clustering algorithms, e.g., the average-linkage method [Jain and Dubes 1988] and the Ward's method [Jain and Dubes 1988]. These algorithms primarily differ in how they update the similarity (or distance) between the existing clusters and the merged clusters.

However, hierarchical clustering algorithms have shortcomings. In hierarchical clustering once a group of objects is merged or split, the process at the next step will operate on the newly generated clusters. Thus merge or split decisions, if not well chosen at some steps, may lead to low-quality clusters. Hierarchical clustering algorithms have been integrated with distance-based iterative relocation or other non-hierarchical clustering algorithms in order to improve the quality of the clusters, e.g., BIRCH [Zhang et al. 1996], CURE [Guha et al. 1998], and Chameleon [Karypis et al. 1999].

2.2.2 Combining Information about Web Contents and Hyperlinks in Search

A user query can be represented as a feature vector. Features, which do not occur in the query, are implicitly assigned zero weights. Individual weights can be assigned to indicate the relative importance of each feature. Let$Q=(q_1,...,q_T)$ be a feature vector of the query.

TF-IDF weights have been widely used for matching a query with Web pages [Hand et al. 2001]. TF stands for feature frequency and simply means that the weight of each feature vector is the frequency by which the feature occurs in the page. This has the effect of increasing the weight on features that occur frequently in a given page. The document-feature matrix in Table 2.2 is expressed in TF form.

However, if a feature occurs frequently in many pages in the page set, then using TF weights for retrieval may have little discriminative power [Hand et al. 2001]. The inverse-document-frequency (IDF) weights help to improve discrimination. For a collection of N pages, the IDF weight of feature j is defined as $\log(N/n_j)$, i.e., the log of the inverse of the fraction of pages in the collection that contain feature$j$. The IDF weight favors features that occur in relatively few pages, i.e., it has discriminative power. The TF-IDF weight is the product of TF and IDF for a feature in a page.

We represent a Web page using TF-IDF weights. A query is expressed in Boolean weights, where 1s for features occurring in the query and 0s otherwise. The relevance-based ranking of a page in terms of distance to the query is the cosine distance between the two feature vectors. The relevance-based ranking of a page can be combined with its authority-based ranking such as PageRank or PageRate as the overall ranking of the page. Ding and Chi [2000] proposed to combine relevance-based rankings, link-based rankings, and usage-based rankings of pages in Web search. In chapter 5, we propose to combine PageRates of pages with their relevance-based rankings. Given a user query$Q$ and a Web page D.

$$R(D,Q) = a \cdot RR(D,Q) + b \cdot AR(D,Q) \tag{2.8}$$

where $R(D,Q)$ is the overall ranking of page$D$, RR is the relevance-based ranking of page D, AR is the authority-based ranking of page$D$, and a, b are weights indicating the importance of the relevance-based ranking and authority-based ranking respectively.

Ding and Chi [2000] classified user queries into exact queries and general queries. In an exact query, users know exactly what they want, so they can use many keywords to describe desired information. The relevance-based ranking should take a large portion in the overall ranking, while the link and link usage based ranking takes a relatively small portion. In a general query, users usually do not have much knowledge on the query topic, so they only use very few keywords and are more likely to see the pages other people reviewed and judged to be relevant. So link and link usage based ranking should be more important in this case than the former case. For these two kinds of queries, different values of$a$ and b are assigned. Users can specify the kind of query explicitly. Some heuristics can also be used to infer the kind of query. For instance, if the number of query keywords is above a threshold, it is considered to be an exact query. Otherwise it is considered to be a general query. Intuitively it is true because users could describe more when they have more knowledge about searched topic.

2.2.3 Classifying and Describing Web Pages

Anchor texts and extended anchor texts have been shown more useful than or in some cases at least as useful as Web page contents in classifying Web pages. Fürnkranz [1999] argued that it is easier to classify a Web page using information describing the links that point to it than using information provided in it. He stressed that using anchor texts provided by multiple authors is less sensitive in classification than having to rely on the page itself by a particular author. His experimental results showed better classification of Web pages using anchor texts than using Web pages themselves. Glover et al. [2002a] used page contents, anchor texts, and extended anchor texts respectively for classifying Web pages. In page contents based classification, each page is represented as a set of features extracted from the contents of the page for classification. In anchor text or extended anchor text based classification, a set of features is extracted from each anchor text or extended anchor text of a page. The sets of features extracted from up to 20 anchor texts or extended anchor texts of a page are aggregated together to get a set of features to represent the page for classification. Glover et al.'s experimental results showed that anchor text based classification is comparable with page contents based classification and extended anchor text based classification is more accurate than both of them.

Since anchor texts and extended anchor texts are collective opinions about the contents of a page, they can be less biased and more conclusive descriptions of the page than the page itself. Anchor texts and extended anchor texts have also been shown useful in describing Web pages and clusters of Web pagesGlover et al. [2002a and 2002b] claimed that the extended anchor texts of a page better summarize the contents of the page since people providing them are interested in the page. Hodgson [2001] showed that anchor texts of a Web page are accurate conceptual descriptions of the contents of the page. Glover et al. [2002a and 2002b] used features extracted from page contents, anchor texts, and extended anchor texts respectively to describe clusters. Their experimental results showed that extended anchor texts have greater descriptive power than both page contents and anchor texts in describing clusters.

In chapter 3, we use extended anchor texts to synthesize the titles of pages and clusters. Glover et al. [2002a and 2002b] treated the feature vector from each extended anchor text of a page equally in aggregating a feature vector for the page. We instead weight the feature vector from each extended anchor text of a page by the in-link strength in aggregating a feature vector for the page. For each cluster, we aggregate the feature vector of each of its member pages to get an aggregated feature vector to describe it.

2.3 Utilizing Information about Web Usage

Information about Web usage has been used to help users navigate the Web. First, useful information can be extracted from Web usage data about the Web and its users. Second, Web usage data are used to construct Markov chain models that can be used for predicting user actions on the Web. Third, Web usage data are used to construct group user models and adaptive Web sites.

2.3.1 Pre-processing Web Usage Data on Web Sites

Web log files are the major sources of Web usage data on Web sites. A Web log file is a collection of records of user requests of documents on a Web server.

An *ECLF (Extended Common Log File) format log file* [Hallam-Baker and Behlendorf 1996] contains records of requests for documents on a Web site. In particular, a record contains eight fields: 1. IP (Internet Protocol) address of the computer from which the request was made; 2. User ID (Identification); 3. Date and time of the request. 4; URL of the requested document; 5. Status indicating whether the request was successful; 6. Size of the document transferred; 7. Referring URL, i.e., the URL of the Web page that contains the hyperlink to the requested document; and 8. Name and version of the browser and operating system being used for making the request. Figure 2.3 shows a record in an ECLF format log file.

```
177.21.3.4 - - [04/Apr/1999:00:01:11 +0100] "GET /studaffairs/ccampus.html HTTP/1.1
200 5327 "http://www.ulst.ac.uk/studaffairs/accomm.html" "Mozilla/4.0 (compatible; MSIE
4.01; Windows 95)"
```

Figure 2.3: A record in an ECLF format log file.

Each time when a user clicks a hyperlink on a page which links to a document on a Web site, if the document can not be found in the user side cache or proxy server cache, the request of the document will be sent to the Web server. When the Web server receives the request, the status of the request is obtained by checking the user's right of access to the document and availability of the document. The status, together with other fields, is put into one record in the Web log file. If the requested page has embedded objects, e.g., graphical, video, audio, or PDF files, a request of each embedded object is automatically made to the Web server. Generally these records of embedded objects are discarded in Web usage mining.

A Web log file can be used to reconstruct user navigation sequences on a Web site. Ideally, we can treat requests having the same IP address as ones from the same user. Since a user may have multiple goals during a visit to the Web site, a user sequence may consist of a number of user sessions. A user session is usually defined as a sequence of requests from the same IP address such that no consecutive requests are separated by more than X minutes, e.g., X =30 minutes [Borges 2000].

However, due to the influence of caching on the user side and proxy server, not all the requests are recorded in Web log files. Multiple users behind the same proxy server may visit a Web site at the same time, thus all their requests seem to come from the same IP address. Some ISPs (Internet Service Providers) dynamically allocate different IP address to a user during a user session [Pirolli and Pitkow 1999]. Thus the accuracy of user sequences and sessions inferred from Web log files can be severely affected [Borges 2000]. *Cookies* can help track individual users within a Web site [Pirolli and Pitkow 1999]. However, due to privacy reasons most users are reluctant to accept cookies [Pirolli and Pitkow 1999; Kobsa and Schreck 2003].

During a user's visit to a Web site, some Web pages may be revisited for ease of navigation, e.g., a user presses the "Back" button to go back to a visited page. In order to concentrate on the meaningful part of a user sequence, Chen et al. [1998] proposed a maximal forward path method to filter out backward links in user sequences for their maximal forward paths.

2.3.2 Utilizing Markov Chain Models for Navigation

Markov chains have been widely used to model user navigation on the Web. Web pages can be treated as states, and hyperlinks between Web pages as one-step transitions between these states in a Markov chain model. Information about Web usage contained in a Web link structure can be used to infer the transition probabilities between these states. The Markov chain model can be used to predict the Web pages that a user is likely to visit given a sequence of Web pages already visited by the user. Link prediction can assist users to find desired information on the Web effectively and efficiently.

2.3.2.1 Basics of Markov Chain Models

A *stochastic process* is a discrete-time process consisting of a sequence of events $\{X_n\}$, where n=0, 1, 2, … identifies the steps in the sequence, and X_n can take any value from a state space $S=\{1, \ldots, m\}$. The outcome of the next event is dependent on the current and previous ones in the sequence, i.e., X_0, X_1, …, X_{n-1} are the past events, X_n is the current event, and X_{n+1} is the future event, and the probability of $X_{n+1}=j$ given $X_0=i_0$, $X_1=i_1,\ldots,X_n=i_n$ is $P\{X_{n+1}=j\}=P\{X_{n+1}=j \mid X_0=i_0, X_1=i_1,\ldots,X_n=i_n\}$. Calculating $P\{X_{n+1}=j\}$ requires the complete information about the outcomes of the process from the initial step to the current step.

However, in many situations, the influence of the earlier outcomes of the process on its future one tends to diminish rapidly as time passes. Thus we can assume that X_{n+1} is dependent only on i previous outcomes, where $i \geq 1$ is a fixed and finite number. In this case, obtaining $P\{X_{n+1}=j\}$ requires only the information about the previous i outcomes, i.e., $P\{X_{n+1}=j \mid X_0=i_0,\ldots,X_n=i_n\}=P\{X_{n+1}=j \mid X_{n-i+1}=i_{n-i+1},\ldots,X_n=i_n\}$. We call this discrete-time sequence of events an *i th-order discrete-time Markov chain*, i.e., a *Markov chain* is a simple form of a stochastic process where the probability of an outcome depends only on a fixed number of previous outcomes. When the next outcome depends on none of the earlier ones, the chain is a sequence of independent events. When the next outcome depends on only the current event, the chain is a *first-order Markov chain*. First-order Markov chains are widely used and high-order Markov chains for $i \geq 2$ can be transformed to first-order Markov chains [Minh 2000]. Unless otherwise specified, we usually refer to a first-order Markov chain simply as a Markov chain.

Since the behavior of a Markov chain at time $n+1$ is dependent only on its behavior at time n, the chain can be described completely by its one-step transition probabilities at step n as $P_{ij}(n) = P\{X_{n+1} = j \mid X_n = i\}$ for all $i, j \in S$, and $n \geq 0$. These probabilities can be grouped together into a transition matrix as $\mathrm{P} = \{ P_{ij}(n) \}_{i,j \in S}$. Since the process must visit some state in state space S at each step, the sum of transition probabilities in each row must be 1, i.e., $\sum_{j \in S} P_{ij}(n) = 1$ for all $j \in S$ and $n \geq 0$. We define a *Markov chain model* as a three-tuple (S, P, $\pi^{(0)}$), where $\pi^{(0)}$ is a vector containing the *initial probability distribution* of X_0 in state space S.

Given $\pi^{(0)}$ as the initial probability of being in each state, we can predict the one-step probability distribution of X_1 as $\pi^{(1)} = \pi^{(0)} \cdot \mathrm{P}$. We can further predict the n th-step probability distribution of X_n as $\pi^{(n)} = \pi^{(n-1)} \cdot \mathrm{P} = \pi^{(n-2)} \cdot \mathrm{P} \cdot \mathrm{P} = \ldots = \pi^{(0)} \cdot \mathrm{P}^{(n)}$. We assume that a Markov chain is *homogeneous*, i.e., the transition probabilities are not dependent on the time n, $P_{ij}(n) = P_{ij}$ for all $i, j \in S$ and $n \geq 0$.

A *path of length k* in a Markov chain is a sequence of states a user visits from step n to step $n+k$. For a first-order Markov chain $\{X_n\}$, n =0, 1, 2,..., we get

$$\begin{aligned}
&P\{X_{n+k} = i_{n+k}, \ldots, X_{n+1} = i_{n+1}, X_n = i_n\} \\
&= P\{X_{n+k} = i_{n+k} \mid X_{n+k-1} = i_{n+k-1}\} \cdot \ldots \cdot P\{X_{n+1} = i_{n+1} \mid X_n = i_n\} \cdot P\{X_n = i_n\} \\
&= P_{i_{n+k-1}, j_{n+k}} \cdot \ldots \cdot P_{i_n, j_{n+1}} \cdot P\{X_n = i_n\}
\end{aligned}$$

The probability of a transition in k steps from state i to state j or $P\{X_{n+k} = j \mid X_n = i\}$ denoted by $P_{ij}^{(k)}$ is given by $\mathrm{P}_{ij}^{\ k}$. Intuitively, this probability may be calculated as the summation of the transition probabilities over all possible k -step paths between i and j in a graph that is equivalent to the transition matrix P, with the transition probability along any path being the product of all successive one-step transition probabilities. This is stated precisely by the *Chapman-Kolmogorov equation* [Ross 1983] as follows.

$$P_{ij}^{(m+n)} = \sum_h P_{ih}^{\ (m)} P_{hj}^{\ (n)} = \mathrm{P}_{ij}^{\ (m+n)} \tag{2.9}$$

where (m,n)=1, 2, …, $\mathrm{P}^{(m+n)}$=$(P_{ij}^{(m+n)})_{i,j\in S}$ is called the $m+n$th-step transition matrix, and $\mathrm{P}_{ij}^{(m+n)}$ is equal to the (i,j)-element of $\mathrm{P}^{(m+n)}$. $\mathrm{P}^{(m+n)}$=P^{m+n}, i.e., $m+n$th-step transition matrix is the $m+n$th power of the one-step transition matrix.

The *total number of visits* that a Markov chain makes to state j in the first n steps, from step 1 to step n, starting from state i is $v_{ij}^{(n)}$=$(v^{(n)})_{ij}$=$(\sum_{k=1}^{n}\mathrm{P}^k)_{ij}$ [Minh 2001].

Given the initial probability distribution $\pi^{(0)}$, the probability distribution at the n th-step is $\pi^{(n)}$=$\pi^{(0)}\cdot\mathrm{P}^n$ for all $n \geq 0$. When $n \to \infty$, for some Markov chains, $\pi^{(n)}$ converges to a fixed vector π, and we can get $\pi\cdot\mathrm{P}$=π. π is called a *stationary distribution* of the Markov chain. π is also the *eigenvector* of the transition matrix P [Minh 2001]. A Markov chain is called *irreducible* if for all pairs of states $(i,j)\in S^2$ there exists an integer n such that $(\mathrm{P}^n)_{ij}$>0, i.e., there is at least one path from state i to state j.

Spears [1998] proposed a transition matrix compression algorithm, which aggregates states in a Markov chain model having similar transition behaviors into a new state. Transition probabilities between the new state and the other states are obtained from the transition probabilities between the aggregated states in the new state and the other states. The loss of transition information of the Markov chain model can be measured by the difference between the k th power of the compressed transition matrix P_c and the compressed k th power of the original transition matrix P, i.e., $(P_c)^k - (P^k)_c$.

In chapter 4, in link prediction on a Web site link structure, we need to raise the transition matrix P to its k th power. For a large P, this is computationally expensive. Spears' algorithm [Spears 1998] can be used to compress the original P to a much smaller matrix P_c without having a significant number of errors since the accuracy experiments on large matrices have shown that $(P_c)^k$ and $(P^k)_c$ are very close to each other. Since the computational complexity of P^k is $O(N^3)$ [Spears 1998], by dramatically reducing the number of states, the time taken by compression can be compensated by all the subsequent probability computations.

2.3.2.2 First-Order Markov Chain based Link Prediction

Bestavros [1995] and Zukerman et al. [1999] suggested using Markov chain models to predict user requests on the Web under the collaborative approach, which assumes that a user behaves in a similar way as other users. A Markov chain model can be built using data collected from a group of users, and then used to make predictions about a new user in the absence of information about the user. When a user visits a Web site, a Markov chain model uses its information regarding the habits of all past users in order to predict the pages the user is most likely to request next.

In chapter 4, we follow the collaborative approach to build Markov chain models for link prediction. We use user link traversals as their collective behavior in using these links to build a Markov chain model. Under the assumption that an individual user behaves similarly to the group of users, the Markov chain model is used for link prediction.

We use user link traversals to estimate transition probabilities between states of a Markov chain model. The transition probability between two states measures the level of relatedness between them. Other kinds of methods have also been used to measure the level of relatedness between documents in a Markov chain model.

Bestavros [1995] considered two documents to be related if users have accessed them in the past within a certain time interval. Padmanabhan and Mogul [1996] considered relatedness between two documents as the probability that one will be accessed by users soon after the other. Whereas Bestavros [1995] defined “soon” by an amount of time, Padmanabhan and Mogul [1996] defined it by a number of accesses from users. Danilowicz and Balinski [2001] used Markov chain models to rank documents in the answer set returned to users. In the Markov chain model, documents are the states, and one-step transition probabilities are proportional to content-based similarities between these documents.

In using Markov chain models for link prediction, different methods have been used to calculate the probabilities of visiting other pages given already visited pages.

Given a sequence of pages visited by a user, Sarukkai [2000] proposed a variant of the first-order Markov chain to accommodate weighting of more than one previously visited page in predicting the most probable next step by the user. His experiments showed that the next step can be predicted correctly with the highest probability over 60% of the time, and over 70% of the pages actually visited by users are in the top 20 predicted pages for the next step.

In chapter 4, we extend Sarukkai’s [2000] work by predicting the most probably to-be-visited (MPT) pages by a user within the next n steps given a sequence of previously visited pages by the user. Generally, pages visited more recently have more influence in predicting the future, and link prediction for the nearer future (when n is small) is more accurate than link prediction for the farther future (when n is large). We use weights to reflect the different levels of influence in predicting most probably to-be-visited pages within the next n steps. By taking into account more steps, our method can provide more insights into the future and the prediction results are more conclusive than just considering the next step.

Sarukkai [2000] also suggested *tour generation* to predict a sequence of pages a user is most likely to visit next. All the unvisited pages are computed for probabilities to be visited using the Markov chain model. They are then picked by the maximal URL path prefix matching with the start state (the current page) of the user. In chapter 4, we construct a Markov chain model of a link hierarchy for link prediction. The Markov chain model is used for predicting guided paths on the link hierarchy. Given a user’s current page on a certain level, the probabilities of visiting pages on the adjacent lower levels down to the lowest level of the link hierarchy are calculated respectively. Pages with the highest probabilities on each level are concatenated with each other to form guided paths.

Markov chain models have been used to solve various navigation problems. Anderson et al. [2001] proposed the MINPATH algorithm, which finds shortcut links for mobile users to improve their wireless Web navigation. MINPATH finds shortcuts by using a first-order Markov chain model of Web visitor behavior to estimate the savings of shortcut links, and suggest only the few best links. They assumed that wireless visitor behavior is dominated by information gathering tasks, which can be accomplished by viewing specific destination pages on the site. A Markov chain model of Web usage is learned from Web log files offline. The objective is to provide shortcut links to visitors in order to shorten long navigation trails. The savings that a single shortcut$p \rightarrow q$ offers is the number of links the visitor can avoid by following the shortcut. Starting at the current page by a user, p, they compute the probability of following each link in$p$ and recursively traverses the Web site link structure until the probability of viewing a page falls below a threshold, or a depth bound is exceeded. The savings of each page is the product of the probability of reaching that page along a path from the current page and the number of links saved. The best$m$ shortcuts are returned. Their experiments showed that a first-order Markov chain model outperforms a zero-order Markov chain model and Naïve Bayes model.

Cadez et al. [2003] proposed a method for the visualization of user Web navigation patterns. A model based clustering approach is used in which users having similar navigation patterns are grouped into the same cluster. The user behavior within each cluster is represented as a Markov chain model.

The stationary distribution of a Markov chain model can be used for authority-based rankings of Web pages. A Markov chain can be used to model a random walk on a directed weighted graph (DWG). The DWG has the states as nodes, one-step transitions as directed edges between the nodes, and weights on the edges as one-step transition probabilities [Lovász 1996]. The basic idea of PageRank [Brin and Page 1998] is that a random Web surfer at each step is at a Web page$i$, and decides which page to visit on the next step. Given the total number of pages$N$, the number of hyperlinks in page$i$, $Num\_link(i)$, and a dampen factor d, with equal probability $d / Num\_link(P)$ the surfer follows a hyperlink to another page, and with equal probability $(1-d)/N$ the surfer jumps to any other page. The transition matrix P is $d \cdot U + (1-d) \cdot M$, where U is the transition matrix of uniform transition probabilities ($U_{ij} = 1/N$ for all i, j), and M is a transition matrix normalized from the adjacency matrix of the Web link structure. The vector of PageRanks, R, is defined as the stationary distribution of a Markov chain model. Equivalently, R is the eigenvector of the transition matrix P.

2.3.2.3 Higher-Order Markov Chain based Link Prediction

User sequences can be reconstructed from Web usage data such as Web log files for building higher-order Markov chain models. In user sequences consisting of ordered pages, if page P frequently follows the same set of ordered pages $(P_1,\ldots,P_k)$, a sub-sequence $(P_1,\ldots,P_k,P)$ of user sequences is frequent. An *association rule* can be formulated as $(P_1,\ldots,P_k) \rightarrow P$, where $(P_1,\ldots,P_k)$ are the conditions and P is the consequence. *Support* of the rule [Han and Kamber 2001] is proportional to the frequency of the sub-sequence as $Num(P_1,\ldots,P_k,\ P)$, and *confidence* of the rule [Han and Kamber 2001] is the number of sub-sequence $(P_1,\ldots,P_k,\ P)$ divided by the total number of sub-sequences having $(P_1,\ldots,P_k)$ as prefix, i.e., $Num(P_1,\ldots,P_k,P)/Num(P_1,\ldots,P_k,X)$, where X is any page following P_k in a sub-sequence. User sequences are identified from Web log files and are used for constructing association rules with their supports and confidences above setting thresholds. These association rules can be used to build a *k th-order Markov chain model*, where k is the number of pages used as conditions in the association rules.

Yang et al. [2001] used k th-order Markov chain models for improving performance of caching proxy servers. Pitkow and Pirolli [1999] proposed *all-k th-order Markov chain models*, where k can take various values. High-order, i.e., frequent long sub-sequences by users, is used for more accurate prediction whenever possible. They also proposed *all-k th-order LRS* (Longest Repeating Sub-sequence) models to reduce the size of all-k th-order Markov chain models while maintaining their prediction accuracy. They proposed an algorithm to mine the longest repeating sub-sequences (LRSs) from

Web log files. Their experiments showed that all$k$ th-order LRS models have almost the same prediction accuracy as all-k th-order Markov chain models while reducing the complexity by over an order of magnitude. Borges and Levene [1999] proposed a HPG (Hypertext Probabilistic Grammar) model based on$k$ th-order Markov chain models to characterize user Web navigation patterns. User navigation sessions are reconstructed from Web log files in building the HPG model. They used entropy to measure the statistical properties of a HPG model.

Pirolli and Pitkow [1999] presented a study of the quality of$k$ th-order Markov chain models ($k \geq 1$) for predicting user surfing patterns. Their experiments, in which ten days of Web log files collected at the Xerox.com Web site were used, showed that a high-order Markov chain model is more accurate than a low-order one. However, the largest reduction in uncertainty is achieved when moving from a zero-order model to a first-order model. The model accuracy is measured using the information theoretic measure of conditional entropy. Pirolli and Pitkow showed that the model probabilities are more stable over time for low-order Markov chain models than for high-order ones. They also presented several methods for user sequence reconstruction from Web log files. They concluded that user sequences could not be very reliably constructed due to two major reasons. First, due to the influence of caching, Web pages in user sequences cannot be reliably recorded in Web log files. Second, there is no reliable method for user identification due to the influence of dynamic IP, proxy server shared by multiple users, multiple users sharing the same computer and users' reluctance to accept cookies etc. Thus they suggested that low-order Markov chain models are more suitable for link prediction than high-order ones.

Their work has supported our method of using first-order Markov chain models for link prediction. Using referrers for constructing a first-order Markov chain model, our method is much less susceptible to the caching and user identification problems. The accuracy of link prediction using first-order Markov chain models is also comparable to the accuracy of high-order ones according to Pirolli and Pitkow [1999]. High-order Markov chain models can be used for link prediction when no major caching influence is present and individual users can be reliably identified so that user sequences can be accurately reconstructed from Web log files.

Chen and Cooper [2002] used continuous-time stochastic models, which are based on semi-Markov chains, to derive user state transition patterns in a Web-based information system. User sessions reconstructed from Web log files were categorized into six user groups based on the similarity in their use of the system. They used a three-layer hierarchical taxonomy of the Web pages in the system. User sessions in each user group were transformed into a sequence of states in the hierarchical taxonomy. The models take into account two factors: the probability that a user moves from one state to another, and the time spent in each of the states. They found that all the user groups but one have three-order sequential dependency. Knowledge of the extent of sequential dependency can be used to predict a user's next move in the system based on his/her past moves.

In chapter 3, we propose a novel method to construct a link hierarchy of a Web site from a Web log file. Web pages are put onto multiple conceptual levels of the link hierarchy. We further propose the PageCluster algorithm to cluster Web pages on the same conceptual level into conceptual clusters based on their link similarities. Clusters and unclustered pages are used to construct a conceptual link hierarchy. Link hierarchies and conceptual link hierarchies are constructed automatically while the hierarchical taxonomies [Chen and Cooper 2002] are constructed manually. User sequences are transformed into sequences of states on a hierarchical taxonomy for Markov chain based link prediction [Chen and Cooper 2002]. Similarly, in chapter 4, we transform user sequences into sequences of states on a link hierarchy and conceptual link hierarchy for Markov chain based link prediction. We assume that all user sequences are in one user group and have one-order sequential dependency in state transitions, while user sequences are categorized into six groups and have high-order sequential dependency in state transitions in Chen and Cooper's work.

2.3.3 Building Group User Models and Adaptive Web Sites for Navigation

In some early work, weights are assigned to the links in a Web link structure based on explicit user feedback. Kaplan et al. [1993] developed the HYPERFLEX system that uses such weights to represent collective user preferences. They argued that a hypertext system can be seen as a semantic network and can thus be represented by an*associative* *matrix* whose elements are link weights indicating strengths of relationships between information topics in the network. Link weights are updated according to explicit user feedback.

Web log files have been used to recognize collective user behavior in using the link structure of a Web site for navigation. Bollen and Heylighen [1998] proposed to order the links in a link structure dynamically for navigation. The link weights are updated based on user traversals on the links. The links in the link structure are then ordered dynamically in the descending order of link weights.

Adaptive Web sites can automatically change their presentation and organization to assist user navigation by learning from Web usage data [Perkowitz and Etzioni 1997]. Perkowitz and Etzioni [1998] proposed the*PageGather* algorithm to find clusters of Web pages on a Web site, which have been found the most frequently associated with each other during past user visits to the Web site recorded in Web log files. They further proposed the *SCML* algorithm [Perkowitz and Etzioni 1999], which uses a conceptual learning algorithm to find a conceptual cluster from each of the clusters found in PageGather and a concept to describe it.

There are three differences between Perkowitz and Etzioni's algorithms and our PageCluster algorithm. First, PageCluster clusters Web pages based on the in-link and out-link similarities of Web pages. Second, PageGather could potentially flatten the structure of the Web site by grouping pages from across the Web site onto a single index page. Therefore, important structural and hierarchical information may be lost. PageCluster instead clusters conceptually related Web pages on each conceptual level of the link hierarchy. Third, Web pages in each of the clusters that PageCluster generates are already conceptually related. PageCluster therefore only needs to synthesize the title of each cluster.

2.4 Summary

Due to the sheer size, still expanding, and heterogeneous nature of the Web, how to navigate and search for desired information on behalf of users has been posed as a major challenge to Artificial Intelligence research [Etzioni 1996]. The Web can be seen as a huge link structure containing three types of information ready for mining, that is, information about hyperlinks, information about Web contents, and information about Web usage. Perkowitz and Etzioni [1997] challenged the AI community to mine useful knowledge about a Web site and its users in order to help users find information on the Web site more easily. In response to their challenge, we have carried out in this thesis the study of building an adaptive Web site by mining the information about the Web site and its users for a number of reasons. First, information about a Web site and its user is easily accessible with relatively uniform quality. In particular, we use Web log files that are rich sources of usage data. Second, a Web site usually has a consistent organization developed by a Web site designer. Third, Web pages on a Web site are of relatively uniform quality. Fourth, changes to the organization and presentation of a Web site can be easily carried out. Fifth, it is possible to evaluate the effects of changes made to a Web site using Web log files or user side agents that collect usage information.

Chapter 3

MINING LINK HIERARCHIES AND CONCEPTUAL LINK HIERARCHIES FOR ADAPTIVE WEB SITE NAVIGATION

In this chapter, we present our approaches to mining link hierarchies and conceptual link hierarchies from Web log files for adaptive Web site navigation. Link hierarchies and conceptual link hierarchies reflect users' collective view in using the link structures of Web sites. Link hierarchies and conceptual link hierarchies can be visualized for helping users find desired information on Web sites effectively and efficiently. The chapter is organized as follows. In section 3.1, we discuss the problems this chapter addresses. In section 3.2, we give an overview of our approaches. In section 3.3, we present a novel method for constructing the link hierarchy of a Web site based on user traversals on hyperlinks. In section 3.4, a hierarchical clustering algorithm called PageCluster is presented to find conceptual clusters on each conceptual level of a link hierarchy and construct a conceptual link hierarchy of the Web site for navigation. In section 3.5 we compare our approaches with related work. Finally we conclude in section 3.6.

3.1 Motivation

With the ever-expanding WWW, Web sites are getting increasingly complicated. It has become very difficult for users to find desired information on large Web sites. How to help users find desired information effectively and efficiently is crucial to the success of a Web site.

Nielsen [2000] identified three fundamental questions that users might ask when they navigate a Web site, namely,*Where am I now? Where have I been? Where can I go next?* Nielsen emphasized the importance of structural navigation in the new generation of Web browsers. The Web site structure should be visualized into different levels for users to know their current locations relative to the Web site as a whole. The visualized site structure helps users understand the relationships between the pages they have visited. Users can decide where they can go next by understanding their current locations and the pages they have visited. By visualizing the site structure, it has been made easier for users to answer the above three navigation questions. Most Web sites may have an underlying hierarchical organization. However, the hierarchy is often buried under a maze-like link structure.

In this chapter, we propose our approaches to visualizing the link structure of a Web site in a link hierarchy and conceptual link hierarchy to help solve the navigation problem. User traversals on hyperlinks between Web pages can reveal semantic relationships between these pages. We use user traversals on hyperlinks as weights to measure semantic relationships between Web pages. On the basis of these weights, we propose a novel method to put Web pages on a Web site onto different conceptual levels in a link hierarchy. We develop a clustering algorithm called*PageCluster*, which clusters conceptually related pages on each conceptual level of the link hierarchy based on their in-link and out-link similarities. Clusters are then used to construct a conceptual link hierarchy, which is visualized to help users navigate the Web site. Our work also presents new approaches to building adaptive Web sites, which can automatically change their organization and presentation to assist user navigation by learning from Web usage data [Perkowitz and Etzioni 1997].

Web sites with a hierarchical organization of Web pages are convenient for both Web site designers to organize information and users to navigate among the Web pages that have been divided into different conceptual levels [Rosenfeld and Morville 1998; Farkas and Farkas 2000; Nielsen 2000]. Users enter a Web site, go through multiple conceptual levels of the Web site, and find desired information in one or more pages. The more links users need to travel through to view a Web page, the less visits the Web page receives [Huberman et al. 1998].

The breadth-first search method is commonly used to construct the link hierarchy of a Web site, in which each page is put onto a conceptual level of the link hierarchy determined by the shortest path from the home page to the page. The distance between the two pages is measured in terms of the number of links between them. The shortest weighted path method has recently been used in a system called MAPA to extract a hierarchical structure from an arbitrary Web site for navigation [Durand and Kahn 1998], in which each page is put onto a conceptual level of the link hierarchy determined by the "minimum-weight"*path*. In MAPA, the "minimum-weight"*paths* are calculated on the basis of heuristically-determined and user-assigned weights on hyperlinks between Web pages to reflect the conceptual relatedness between these pages. We propose a novel method for link hierarchy construction that uses user traversal information. Among all links to a Web page, the link that has been the most often traversed is identified as the *main link* of the page. Pages are put onto different conceptual levels of the link hierarchy determined by their main links.

Bibliographic analysis based on the notions of co-citation and coupling has been used to cluster documents. It has now been extended for clustering Web pages [Almind and Ingwersen 1997; Giles et al. 1998; Davison 2000; Henzinger 2000 and 2001]. We propose the notion of link similarity, including in-link and out-link similarities between Web pages. We develop a clustering algorithm called PageCluster based on in-link and out-link similarities to cluster pages on each conceptual level of the link hierarchy to form two kinds of clusters, namely, navigation and category clusters.

The clusters generated by PageCluster are used to construct a conceptual link hierarchy of the Web site to reduce page clutter in the link hierarchy. The conceptual link hierarchy is visualized in a prototype called ONE. ONE helps users control navigation by themselves. Users are not confined to following only the hyperlinks in each page. They can understand their current locations in the context of different conceptual levels consisting of pages and clusters in the conceptual link hierarchy. They can move up and down in the conceptual link hierarchy or jump from one page or cluster to another. They can also zoom in a particular cluster to view the pages in it. The conceptual link hierarchy gives users a clearer view of their current locations on the Web site and the relationships between pages and clusters in the hierarchy.

3.2 Overview

As discussed in chapter 2, three major approaches have been proposed to tackle the navigation problem. First, the link structure of a Web site has been used for visualization [Duran and Kahn 1998; Munzner 2000] and clustering [Kleinberg 1998; Kumar et al. 1999; Flake et al. 2002; Larson 1996; Pitkow and Pirolli 1997; Dean and Henzinger 1999]. Second, anchor texts and extended anchor texts have been shown more useful than Web page contents in classification of Web pages [Fürnkranz 1999; Glover et al. 2002a]. Anchor texts and extended anchor texts have also been used to extract the titles of pages and clusters [Glover et al. 2002a and 2002b; Hodgson 2001]. Third, Web usage data, which contains records of how users have visited a Web site, have been used to identify collective user behavior in using the Web site [Bollen and Heylighen 1998; Perkowitz and Etzioni 1998]. The link structure of a Web site has major influence on how users use the Web site since most users mainly follow hyperlinks in navigation. On the other hand, Web usage data can reflect some characteristics of the link structure of a Web site. We have combined all these three approaches in our approach to mining conceptual link hierarchies. First, we propose a method to construct a link hierarchy of a Web site on the basis of user traversals recorded in a Web log file. Second, we develop the PageCluster algorithm to cluster conceptually related pages on each conceptual level of the link hierarchy based on their in-link and out-link similarities. Third, clusters are used to construct a conceptual link hierarchy of the Web site, which is more compact than the link hierarchy. Extended anchor texts are used to synthesize the titles of pages and clusters in the conceptual link hierarchy. Finally, the conceptual link hierarchy is visualized for user navigation.

3.3 Link Hierarchy Construction

In this section, we present a method for constructing the link hierarchy of a Web site from an ECLF (Extended Common Log File) format log file [Hallam-Baker and Behlendorf 1996].

An ECLF format log file contains records of requests for documents on a Web site. In particular, a record contains the URLs of both the requested document and the referrer indicating where the request comes from. In chapter 2, Figure 2.3 shows a record in an ECLF format log file.

The records of embedded objects in Web pages, including graphical, video and audio files etc, are treated as redundant requests and removed, since every request of a Web page automatically initiates a series of requests of all the embedded objects in the page. The records of unsuccessful requests are also discarded as there may be bad links, missing or temporarily inaccessible pages, or unauthorized requests etc. In our approach, only the URLs of the requested Web page and the corresponding referrer are used for link hierarchy construction. We therefore have a set of links $WL = \{(r, u)\}$, where r and u are the URLs of the referrer and the requested page respectively. Since the same link[3] may have been followed by various users in their visits, the links are aggregated to get a set of aggregated links $WL' = \{(r, u, w)\}$, where w is the number of traversals from r to u.

We remove links from WL' whose referrers are outside the current Web site. We construct an *adjacency matrix* Q to represent all the links in WL'. The columns and rows of Q are indexed by all the unique pages in the requested pages and referrers[4]. For each entry in Q indexed by row i and column j, if there is a link from i to j, we put the number of traversals on the link into the entry. Otherwise, we put zero into the entry.

In a hierarchical Web site, Web pages are organized onto different conceptual levels. The home page is on the first conceptual level and links to a set of Web pages on the second conceptual level. The home page can be seen as containing a summary of the contents of the pages on the second conceptual level, in the form of either the contents of the home page itself or the extended anchor texts of the links to these pages in the home page. The home page is conceptually more general than the pages on the second conceptual level. Similarly, each page on the second conceptual level can be seen as containing a summary of the more specific contents of the pages on the third conceptual level. This in general applies to any two adjacent conceptual levels of a hierarchical Web

[3] In most cases a link is the hyperlink from r to u. When “-“ is in the referrer field, the URL of the referrer is unavailable. We assume that there is a virtual link from “-“ to the requested page.

[4] Pages are uniquely identified by their URLs.

site. The lower a conceptual level is, the more specific the contents of pages on the conceptual level are.

In a hierarchical Web site, the links from pages on a higher conceptual level to pages on its adjacent lower conceptual level define the main structure of the Web site. These links should be distinguished from the other links [Conklin 1987; Parunak 1991]. Farkases [2000] identified two kinds of links in a hierarchical Web site, namely, primary and secondary links. *Primary links* define the hierarchical structure of the Web site. *Secondary links* provide users with more freedom in navigation between Web pages.

Definition 3.1 (Structural and Secondary Links) A structural link is a hyperlink from a page on a higher conceptual level to a page on its adjacent lower conceptual level in the hierarchy of a Web site. All the non-structural links are secondary links.

A link structure consisting of both structural and secondary links is shown in Figure 3.1, where the secondary links have distorted the hierarchical organization of the Web site.

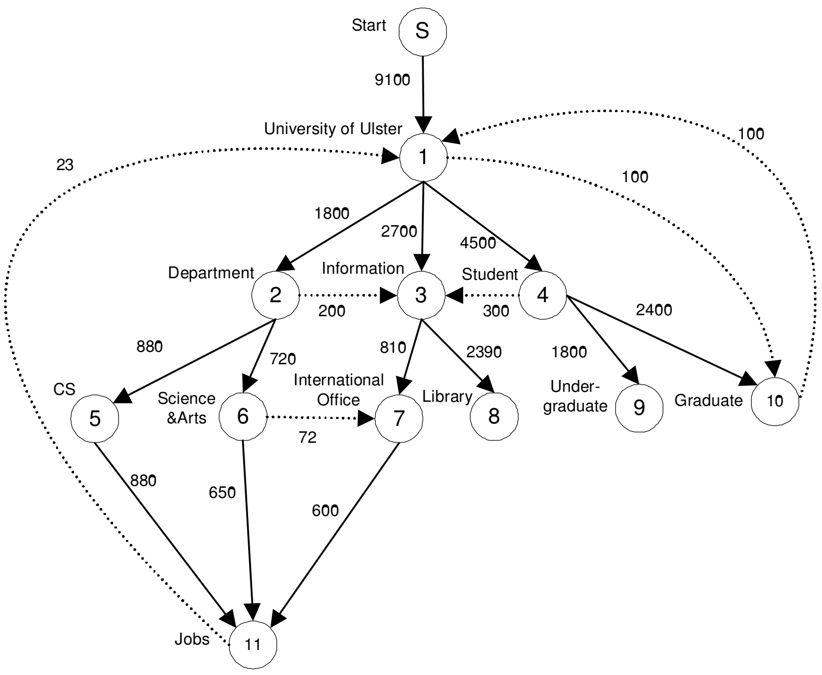


Figure 3.1: The link structure of a Web site consisting of Web pages, structural and secondary links, and numbers of traversals on these links. Structural links are shown in solid lines. Secondary links are shown in dotted lines.

Definition 3.2 (Link Hierarchy) The link hierarchy of a Web site consists of pages on different conceptual levels and structural links between pages on two adjacent conceptual levels.

We now propose a method for constructing the link hierarchy of a Web site. In our method, we treat the number of user traversals on a link as the weight on the link. Among all links to a page, the link with the highest weight can be chosen as the main link to the page and the page where the main link comes from is the *main parent* of the page accordingly. We put a page onto a conceptual level of the link hierarchy determined by its main parent. A page may have multiple pages as candidates for its main parent in the link hierarchy if these pages have an equal highest weight on their links to the page. Whichever of these candidate pages appears first on the link hierarchy is chosen as the main parent of the page.

To construct the link hierarchy of a Web site, we start with the home page to form the first conceptual level, L_1, of the link hierarchy. We select those pages that have the home page as their main parent to form the second conceptual level, L_2, of the link hierarchy. To form the ith conceptual level, L_i, of the link hierarchy, we select those pages, each having a page on conceptual level L_{i-1} as its main parent and having not so far appeared in the link hierarchy. We include structural links and discard secondary links in the link hierarchy. We then get the link hierarchy of the Web site.

Figure 3.2 shows the link hierarchy of a Web site. Some pages can have multiple in-links. For instance, page 11 has in-links from page 5, 6 and.7

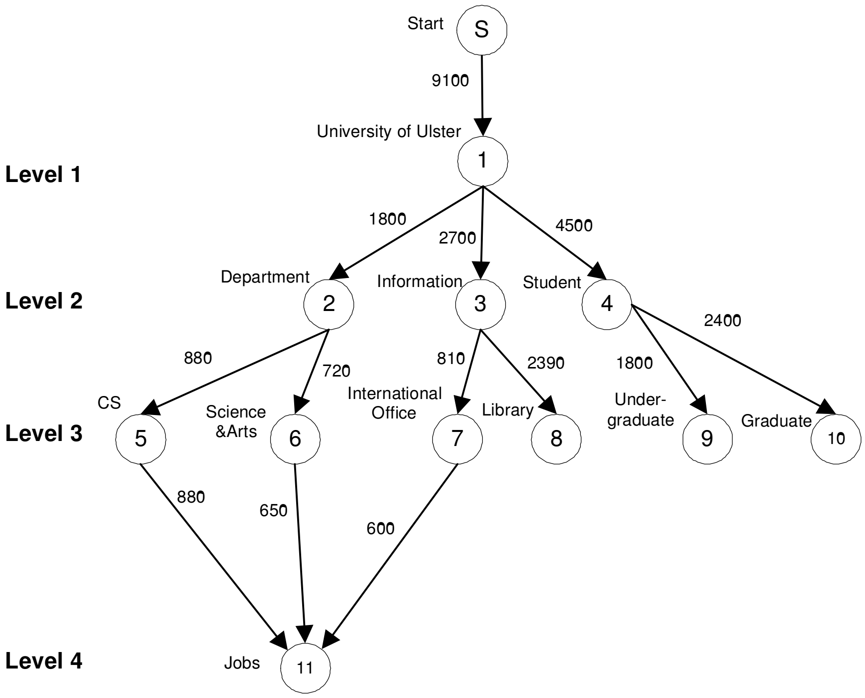


Figure 3.2: The link hierarchy of a Web site constructed from the link structure shown in Figure 3.1.

3.4 Clustering Conceptually Related Pages

The link hierarchy of a Web site can be visualized for navigation. However, there might be still a large number of pages on some conceptual levels of the link hierarchy. To reduce page clutter in the link hierarchy, pages on the same conceptual level need to be clustered.

We propose a hierarchical[5] clustering algorithm called *PageCluster* to cluster conceptually related pages on each conceptual level of the link hierarchy of a Web site based on their in-link and out-link similarities. We then use the clusters to construct a

[5] The clustering algorithm is called hierarchical since every page is initially treated as a singleton cluster, and at each step we merge two of the most similar clusters based on a similarity measure defined between two clusters. The process continues until the similarity between every pair of clusters is above a given threshold.

conceptual link hierarchy of the Web site, which is more compact than the link hierarchy itself. The conceptual link hierarchy can also be visualized for navigation.

3.4.1 Page Clusters and Link Similarity

In this section, we first define two types of clusters that may exist on each conceptual level of the link hierarchy. These two types of clusters are defined on the basis of in-link and out-link similarities respectively. We then describe what we mean by in-link and out-link similarities between two pages and how to measure such similarities.

Definition 3.2 (Category Cluster) A category cluster is a set of Web pages on the same conceptual level of a link hierarchy that are similar in their in-links.

Definition 3.3 (Navigation Cluster) A navigation cluster is a set of Web pages on the same conceptual level of a link hierarchy that are similar in both their in-links and out-links.

We keep structural links in matrix Q and remove secondary links from it to get the *structural link matrix*. We then normalize each row and column of the structural link matrix to get the *out-link* and *in-link strength matrices* respectively.

The structural link, out-link strength, and in-link strength matrices of the link hierarchy in Figure 3.2 are shown in Figure 3.3, 3.4 and 3.5 respectively

Page\Page	1	2	3	4	5	6	7	8	9	10	11
1		1800	2700	4500							
2					880	720					
3							810	2390			
4									1800	2400	
5											880
6											650
7											600
8											
9											
10											
11											

Figure 3.3: The structural link matrix of the link hierarchy shown in Figure 3.2, in which each entry represents the number of traversals on the structural link from the page indexed by the row to the page indexed by the column.

Page\Page	1	2	3	4	5	6	7	8	9	10	11
1		0.2	0.3	0.5							
2					0.55	0.45					
3							0.253	0.747			
4									0.429	0.571	
5											1
6											1
7											1
8											
9											
10											
11											

Figure 3.4: The out-link strength matrix of the link hierarchy shown in Figure 3.2, in which each entry represents the out-link strength of the link from the page indexed by the row to the page indexed by the column.

Page \ *Page*	1	2	3	4	5	6	7	8	9	10	11
1		1	1	1							
2					1	1					
3							1	1			
4									1	1	
5											0.413
6											0.305
7											0.282
8											
9											
10											
11											

Figure 3.5: The in-link strength matrix of the link hierarchy shown in Figure 3.2, in which each entry represents the in-link strength of the link from the page indexed by the row to the page indexed by the column.

Gower [1971] proposed the concept of general similarity coefficient to measure similarity between two cases The concept has been widely used in problems, such as measuring similarity between information items in hypermedia systems [Cunliffe et al. 1997]. Given two cases i and j both having a set of N features, f_l, $1 \le l \le N$, *similarity coefficient* $S_{i,j}$ between i and j is defined as:

$$S_{i,j} = \frac{\sum_{l=1}^{N} w_{i,j,l} \cdot S_{i,j,l}}{\sum_{l=1}^{N} w_{i,j,l}} \qquad (3.1)$$

where $S_{i,j,l}$ represents the similarity between i and j on the lth feature, and $w_{i,j,l}$ is a user-assigned weight indicating the level of importance of the comparison between i and j on the lth feature, but is set to zero when no valid comparison is possible.

Wishart [2001 and 2002] extended the concept of general similarity coefficient, and defined the concept of *Euclidean distance* to measure the distance between two cases based on a set of features. Given two cases i and j, and the values they have on a set of

N features, f_l, $1 \le l \le N$, are $(x_{i,1},...x_{i,N})$ and $(x_{j,1},...x_{j,N})$ respectively, the *Euclidean distance*, $ED_{i,j}$ between i and j is defined as:

$$ED_{i,j} = \left(\frac{\sum_{l=1}^{N} w_{i,j,l} \cdot (x_{i,l} - x_{j,l})^2}{\sum_{l=1}^{N} w_{i,j,l}} \right)^{1/2} \tag{3.2}$$

where $w_{i,j,l}$ is a user-assigned weight indicating the level of importance of the comparison between i and j on the lth feature, but is set to zero when no valid comparison is possible.

We use the concept of Euclidean distance to measure the out-link and in-link similarities of two pages respectively. Common and uncommon out-links and in-links of the two pages are given different weights to reflect their different levels of importance in measuring the out-link and in-link similarities of the two pages respectively.

Definition 3.4 (Out-link Similarity) Given the out-link strength matrix Q_{out} of a link hierarchy, two pages i and j on a conceptual level of the link hierarchy, the ith row in Q_{out}, $\{q_{i,1}, q_{i,2},..., q_{i,N}\}$, and the jth row in Q_{out}, $\{q_{j,1}, q_{j,2},..., q_{j,N}\}$, *out-link similarity* between i and j is defined as a *Euclidean distance*,

$$Out-Sim_{i,j} = \left(\frac{\sum_{l=1}^{N} w_{i,j,l} \cdot (q_{i,l} - q_{j,l})^2}{\sum_{l=1}^{N} w_{i,j,l}} \right)^{1/2}$$

, where $w_{i,j,l}$=0 if $q_{i,l}=q_{j,l}$=0, $w_{i,j,l}=w_C$ if $q_{i,l} \neq 0$ and $q_{j,l} \neq 0$, and $w_{i,j,l}=w_U$ otherwise. w_C and w_U are the weights for the common and uncommon out-links of i and j respectively.

Definition 3.5 (In-link Similarity) Given the in-link strength matrix Q_{in} of a link hierarchy, two pages i and j on a conceptual level of the link hierarchy, the ith column in Q_{in}, $\{q_{1,i}, q_{2,i},..., q_{N,i}\}$ and the jth column in Q_{in}, $\{q_{1,j}, q_{2,j},..., q_{N,j}\}$, *in-link similarity* between i and j is defined as a Euclidean distance,

$$In-Sim_{i,j}=\left(\frac{\sum_{l=1}^{N} w_{i,j,l}\cdot(q_{l,i}-q_{l,j})^2}{\sum_{l=1}^{N} w_{i,j,l}}\right)^{1/2}$$, where $w_{i,j,l}$=0 if $q_{l,i}=q_{l,j}$=0, $w_{i,j,l}=w_C$ if $q_{l,i}\neq 0$ and $q_{l,j}\neq 0$, and $w_{i,j,l}=w_U$ otherwise. w_U and w_C are the weights for the common and uncommon in-links of i and j respectively.

We require $w_U > w_C > 0$, since the common links of two pages indicate a higher level of link similarity than the uncommon links.

3.4.2 PageCluster Algorithm

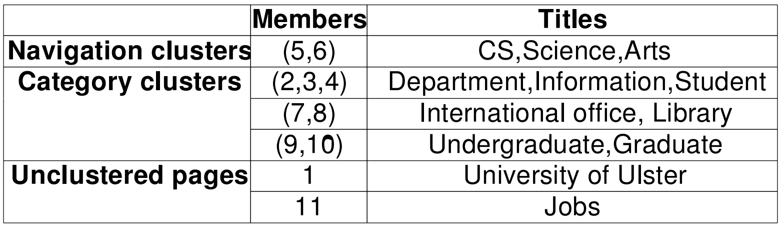
In this section, we present a hierarchical clustering algorithm called PageCluster to cluster pages on each conceptual level of a link hierarchy. PageCluster consists of two algorithms: *navigation clustering* and *category clustering* to find navigation clusters and category clusters respectively. Since navigation clustering takes into account both in-link and out-link similarities while category clustering relies on in-link similarity only, we start with navigation clustering followed by category clustering.

To find navigation clusters, we set two thresholds on in-link and out-link similarities respectively. Merging two clusters requires that both in-link and out-link similarities between every pair of pages from the two clusters are below the given thresholds on in-link and out-link similarities respectively. We use the *complete linkage method*, in which link similarity between two clusters is measured by the link similarity between the two pages from the two clusters, which are the least similar in their links. Link similarity between two page is the average of the in-link and out-link similarities between the two pages. In each step of navigation clustering, two clusters that are the most similar ones in terms of link similarity are merged. When there is more than one pair of the most similar clusters that share a common cluster, the pair of clusters in which there are the most pairs of pages sharing common in-links and out-links are merged. Navigation clustering ends up with a set of navigation clusters and a set of unclustered pages.

To find category clusters, we set a threshold on in-link similarity. Merging two clusters requires that in-link similarity between every pair of pages from the two clusters is below the given threshold on in-link similarity. We use the complete linkage method to cluster unclustered pages left after navigation clustering, where *in-link similarity* between two clusters is measured by the *in-link similarity* between the two pages from the two clusters, which are the least similar in their in-linksIn each step of category clustering, two clusters that are the most similar ones in terms of in-link similarity are merged. When there is more than one pair of the most similar clusters that share a common cluster, the pair of clusters in which there are the most pairs of pages sharing common in-links are merged.

We synthesize titles for pages and clusters using extended anchor texts. Similar to the work by Glover et al. [2002a and 2002b], we extract features from the extended anchor texts of all the pages in the conceptual link hierarchy. All the words and phrases in these extended anchor texts are considered as candidate features. We perform *thresholding*, by removing those rare words and phrases that do not occur frequently in these extended anchor texts. We then get a *feature set* consisting of M features, f_j, $1 \leq j \leq M$. Each extended anchor text E is represented as a feature vector, $E=(w_1, w_2, ..., w_M)$, where $w_j=1$ if E contains feature f_j, and $w_j=0$ otherwise. Some Web pages may have multiple extended anchor texts. Glover et al. [2002a and 2002b] treated the feature vector representing each extended anchor text of a page equally when synthesizing a feature vector to represent the page. Given a page, we use the in-link strength of each link to the page to weight the feature vector of the extended anchor text of the link and add all the weighted feature vectors together to get the feature vector of the page. Features with the highest weights in the feature vector of a page are selected to form the title of the page.

We aggregate the feature vectors of all the member pages of a cluster to get an *aggregated feature vector* to describe the cluster. Features with the highest weights in the aggregated feature vector of a cluster are selected to form the title of the cluster. The clustering results of the link hierarchy in Figure 3.2 are shown in Figure 3.6.



	Members	Titles
Navigation clusters	(5,6)	CS,Science,Arts
Category clusters	(2,3,4)	Department,Information,Student
	(7,8)	International office, Library
	(9,10)	Undergraduate,Graduate
Unclustered pages	1	University of Ulster
	11	Jobs

Figure 3.6: The clustering results of the link hierarchy in Figure 3.2.

3.4.3 Conceptual Link Hierarchies

After pages in a link hierarchy have been clustered, the conceptual link hierarchy of a Web site can be constructed. Clusters and unclustered pages on each conceptual level of the link hierarchy are represented as *nodes* on the same conceptual level of the conceptual link hierarchy. We create a *virtual link* from a node on a higher conceptual level of the conceptual link hierarchy to a node on the adjacent lower conceptual level of the hierarchy if one of the two nodes represents a cluster and one node is a *parent* of the other. We choose those pages, which have the highest aggregated in-link strengths of their links to the member pages of a cluster, as the *parents* of the cluster. A cluster is a *parent* of another cluster if a page in the first cluster is a parent of the second cluster. A cluster is a *parent* of an unclustered page if a page in the cluster is a parent of the unclustered page in the link hierarchy. We keep the hyperlink in the link hierarchy between a node on a higher conceptual level to a node on the adjacent lower conceptual level if both nodes represent unclustered pages.

Definition 3.6 (Conceptual Link Hierarchy) The conceptual link hierarchy of a Web site consists of multiple conceptual levels. The first conceptual level consists of a single node representing the home page. Each of the other conceptual levels consists of nodes representing clusters and unclustered pages. There is a *link* from a node on a higher conceptual level to a node on the adjacent lower conceptual level if the first node is a *parent* of the second.

The conceptual link hierarchy constructed from the link hierarchy in Figure 3.2 is shown in Figure 3.7.

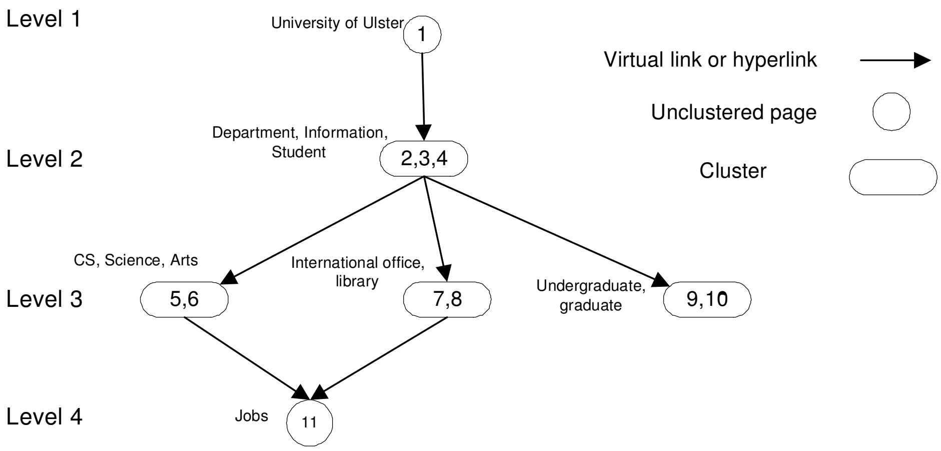


Figure 3.7: The conceptual link hierarchy constructed from the link hierarchy in Figure 3.2.

A cluster on a conceptual link hierarchy can be *zoomed in* to show its member pages and the membership links from the cluster to its member pages. Figure 3.8 shows the conceptual link hierarchy in Figure 3.7, after it has been zoomed in. In the zoomed-in conceptual link hierarchy, new nodes have been created between two adjacent conceptual levels to represent pages in clusters on the higher conceptual level. New virtual links have also been created from pages in clusters on a higher conceptual level to nodes representing clusters on the adjacent lower conceptual level if these pages are the parents of the corresponding nodes. Hyperlinks have also been shown from pages in clusters on a higher conceptual level to unclustered pages on the adjacent lower conceptual level if these pages are the parents of the unclustered pages in the link hierarchy. Virtual links from clusters on a higher conceptual level to nodes on the adjacent lower conceptual level have been hidden away.

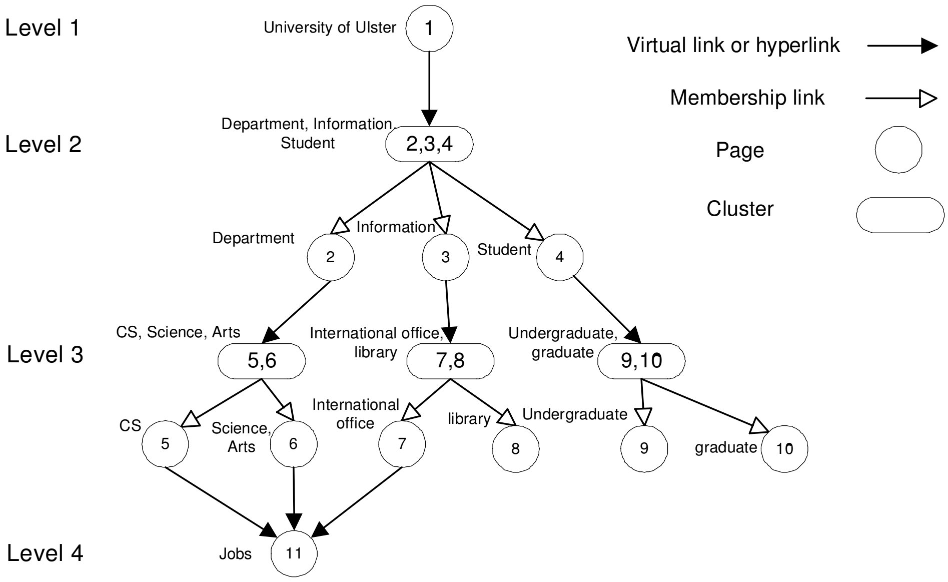


Figure 3.8: The zoom-in form of the conceptual link hierarchy in Figure 3.7.

The conceptual link hierarchy in both forms can be visualized in a prototype ONE for user navigation.

3.5 Comparisons with Related Work

In this section, we compare our approaches with related work. First, we compare our method of constructing link hierarchies with the breadth-first search and shortest weighted path methods. Second, we compare link similarity with bibliographic co-citation and coupling. Third, we compare our method of synthesizing titles of pages and clusters with the method proposed by Glover et al. [2002a and 2002b]. Fourth, we compare the PageCluster algorithm with general hierarchical clustering algorithms. Fifth, we compare the PageCluster algorithm with the PageGather algorithm proposed by Perkowitz and Etzioni [1998].

3.5.1 Comparing Our Method with the Breadth-First Search and Shortest Weighted Path Methods

There are two differences between our method of constructing link hierarchies and the breadth-first search and shortest weighted path methods. First, in our method, we treat the number of user traversals on a link as collective user feedback in using the link and assign it to the link as the weight on the link. The higher the weight on a link between two pages is, the more the two pages are conceptually related In breadth-first search method, all links are assigned equal weights. In shortest weighted path method, we assign weights on links in inverse proportion to the numbers of user traversals on the links. The higher the weight on a link between two pages is, the less the two pages are conceptually related. Second, in our method, among all links to a page, the link with the highest weight can be chosen as the main link to the page and the page where the main link comes from can be the main parent of the page accordingly. Similar to the method proposed by Munzner [2000], we put a page onto a conceptual level of the link hierarchy determined by its main parent. A page may have multiple pages as candidates for its main parent if these pages have an equal highest weight on their links to the page Whichever of these candidates appears on the link hierarchy first is chosen as the main parent of the page. While in both the breadth-first search and shortest weighted path methods, a page is put onto a conceptual level of the link hierarchy determined by the shortest path from the root (home page) to the page.

3.5.2 Comparing Link Similarity with Bibliographic Co-Citation and Coupling

We define link similarity consisting of in-link and out-link similarities instead of co-citation and coupling similarities as the similarity measure for clustering in the PageCluster algorithm. Co-citation and coupling similarities of two Web pages are defined as the number of common in-links and out-links that the two pages have respectively, where each in-link and out-link is treated equally in measuring similarities. We use the numbers of user traversals on the links as collective user feedback in using these links. User traversals on the in-links and out-links of a Web page are used to represent the in-link and out-link strengths of the page respectively. We can interpret link strength as follows. Web masters generally put those links to Web pages, which they think the most relevant to the current page, in the most prominent positions in the current page. These links attract the most users due to their prominence. The more users have traversed these links, the higher the link strengths on these links are. We define the in-link similarity of two pages as a distance-based measure of the link strengths on their common and uncommon in-links. We define the out-link similarity of two pages as a distance-based measure of the link strengths on their common and uncommon out-links. Such defined link similarities can incorporate user preferences and thus are more objective and user-centric.

In-link and out-link similarities are more suitable for Web page clustering than co-citation and coupling similarities. First, since some Web pages have only two or three in-links and out-links, co-citation and coupling similarities are too coarse for clustering and there could be a very large number of pairs of pages having the same similarity measures. Second, in-link and out-link similarities reflect collective user behavior in a given time period. Two pages having link similarity in one period may not be so in another period. Different clusters can be generated to reflect user behavior changes. These changes cannot be reflected in the clusters generated on the basis of co-citation and coupling similarities.

3.5.3 Comparing Our Method with Glover et al.'s Method in Synthesizing Titles for Pages and Clusters

There are two differences between our method and Glover et al.'s method in synthesizing titles for pages and clusters. First, Glover et al. [2002a and 2002b] treated the feature vector representing each extended anchor text of a page equally when synthesizing a feature vector to represent the page. Given a page, we use the in-link strength of each link to the page to weight the feature vector of the extended anchor text of the link and add all the weighted feature vectors together to get the feature vector of the page. Second, Glover et al. [2002a] used features, which can optimally separate a cluster from the rest of the pages measured by expected entropy loss or information gain, to name the cluster. We select features with the highest weights in the feature vector of a page to form the title of the page. We aggregate the feature vectors of all the member pages of a cluster to get an aggregated feature vector to describe the cluster. Features with the highest weights in the aggregated feature vector of a cluster are selected to form the title of the cluster.

3.5.4 Comparing PageCluster with Hierarchical Clustering Algorithms

We present a hierarchical clustering algorithm called PageCluster to cluster pages on each conceptual level of a link hierarchy. We have adapted hierarchical clustering algorithms to the clustering task at hand in three aspects.

First, a hierarchical clustering algorithm generally consists of one clustering process. PageCluster consists of two algorithms: navigation clustering and category clustering to find navigation clusters and category clusters respectively. Since navigation clustering takes into account both in-link and out-link similarities while category clustering relies on in-link similarity only, we start with navigation clustering followed by category clustering, in which we cluster unclustered pages left after navigation clustering.

Second, a hierarchical clustering algorithm generally has one threshold in clustering. To find navigation clusters, we set two thresholds on in-link and out-link similarities respectively. Merging two clusters requires that both in-link and out-link similarities between every pair of pages from the two clusters are below the given thresholds on in-link and out-link similarities respectively. We use the complete linkage method, in which link similarity between two clusters is measured by the link similarity between the two pages from the two clusters, which are the least similar in their links. Link similarity between two pages is the average of the in-link and out-link similarities between the two pages.

Third, in a hierarchical clustering algorithm, two clusters that are the most similar ones in terms of their similarity are merged. When there is more than one pair of the most similar clusters that share a common cluster, the clustering algorithm itself cannot guarantee to make the best merging decision. In each step of navigation clustering, two clusters that are the most similar ones in terms of link similarity are merged. When there is more than one pair of the most similar clusters that share a common cluster, the pair of clusters in which there are the most pairs of pages sharing common in-links and out-links are merged. In each step of category clustering, two clusters that are the most similar ones in terms of in-link similarity are merged. When there is more than one pair of the most similar clusters that share a common cluster, the pair of clusters in which there are the most pairs of pages sharing common in-links are merged.

3.5.5 Comparing PageCluster with PageGather

There are three differences between Perkowitz and Etzioni's algorithms and our PageCluster algorithm. First, PageCluster clusters Web pages based on the in-link and out-link similarities of Web pages. Second, PageGather could potentially flatten the structure of the Web site by grouping pages from across the Web site onto a single index page. Therefore, important structural and hierarchical information may be lost. PageCluster instead clusters conceptually related Web pages on each conceptual level of the link hierarchy. Third, Web pages in each of the clusters that PageCluster generates are already conceptually related. PageCluster therefore only needs to synthesize the title of each cluster.

3.6 Summary

In this chapter, we present a novel approach to the navigation problem. We propose a method to construct the link hierarchy of a Web site using Web log files. We develop the PageCluster algorithm to cluster conceptually related pages on each conceptual level of the link hierarchy based on the in-link and out-link similarities between these pages so that a more compact conceptual link hierarchy of the Web site can be constructed for navigation. Extended anchor texts are used to synthesize the titles of pages and clusters. The conceptual link hierarchy is visualized in a prototype called ONE for user navigation.

User behavior may change over time. Using the Web log files of a Web site taken in different periods of time, we can construct different link hierarchies of the Web site. Consequently, the clustering results on each conceptual level of the link hierarchy and conceptual link hierarchies constructed are also different. In our future work, we intend to study the relationships between the changes of user behavior and the changes in the link hierarchy, clustering results, and the conceptual link hierarchy. By comparing the link hierarchies, clustering results, and conceptual link hierarchies of a Web site in different periods of time, we can investigate some unknown user behavior changes over time. This study also helps answer the question of how to segment a Web log file for link hierarchy construction so that the major user behavior changes can be reflected on the conceptual link hierarchy for user navigation. We also plan to apply our approach to other Web sites for user navigation support.

Chapter 4

LINK PREDICTION FOR ADAPTIVE WEB SITE NAVIGATION

In this chapter, we present our approaches to link prediction using Markov chain models for adaptive Web site navigation. A*Markov chain model* is constructed to reflect a group of users' collective behavior in navigating a Web site. Three types of Markov chain models are constructed using a Web site link structure (MMS), a link hierarchy (MMH), and a conceptual link hierarchy (MMC) of a Web site, respectively. Link prediction results using three types of Markov chain models are used to help users find desired information on the Web site. The chapter is organized as follows. In section 4.1, we discuss the problems this chapter addresses. In section 4.2, we give an overview of our approaches. In section 4.3, we present our methods for constructing Markov chain models from Web site link structures, link hierarchies, and conceptual link hierarchies, respectively. In section 4.4, we use three kinds of Markov chain models for link prediction toward adaptive Web site navigation. In section 4.5, we compare our approaches with related work. Finally we conclude in section 4.6.

4.1 Motivation

Nielsen [2000] identified three fundamental questions that users might ask when they navigate a Web site, namely,*Where am I now? Where have I been? Where can I go next?* In chapter 3, we proposed our approaches to mining link hierarchies and conceptual link hierarchies from Web log files for adaptive Web site navigation. A visualized link hierarchy and conceptual link hierarchy can help users understand their current locations on a Web site and the relationships between the pages they have visited. These understandings help users decide where to find desired information.

In this chapter, we propose our approaches to predicting user navigation on a Web site. Link prediction can help users answer the third navigation question, that is, where I can go next. Predicted pages may be several clicks away from the current page in a link hierarchy or a conceptual link hierarchy. As a result of link prediction, users may take fewer clicks and less time to find desired information contained in the predicted pages than navigating a link hierarchy or conceptual link hierarchy in a normal way. Link prediction is integrated with link hierarchies and conceptual link hierarchies visualized in ONE for adaptive Web site navigation.

We assume that the behavior of an individual user in navigating a Web site can be characterized by the collective navigation behavior of a group of users. We build a model from the collective navigation behavior of a group of users and then use the model to predict an individual user's navigation on a Web site.

User navigation on a Web site can be modeled as a *first-order Markov chain*, i.e., the next page to be visited by a user is only dependant on the current page. User traversals on hyperlinks can be seen as collective opinions of users in following these hyperlinks. We view pages as states and hyperlinks between them as one-step transitions between these states in a *Markov chain model*. User traversals are used to estimate one-step transition probabilities.

The Markov chain model is used to predict pages that a user is most likely to visit in the next step given the current page. Sarukkai [2000] proposed a variation of Markov chains, which predicts the page to be visited in the next step given a sequence of pages visited by a user. He also proposed an approach that generates *guided tours* by successively predicting a sequence of pages on a navigation path.

We propose to construct three kinds of Markov chain models of a Web site, namely, a Markov chain model from a Web site link structure (MMS), a Markov chain model from a link hierarchy (MMH), a Markov chain model from a conceptual link hierarchy (MMC), respectively. There are a number of reasons why three kinds of Markov chain models are needed. First, a link hierarchy includes structural links and excludes secondary links of a Web site link structure. Structural links show conceptual relationships between pages and are traversed most frequently by users in navigating a

Web site. Secondary links have auxiliary navigational functions and are rarely traversed by users in navigating a Web site. By excluding transitions representing secondary links in the transition matrix of a MMH, we expect to predict pages more accurately using MMH than using MMS. Second, conceptually related pages are put into clusters in a conceptual link hierarchy, thus a MMC is more compact than a MMH. Using a MMC is more efficient than using a MMH in link prediction. Conceptual clusters can help users go through a list of prediction results in order to find desired information effectively and efficiently. Third, instead of raising the transition matrixP of a MMH or MMC to its nth power for link prediction, we can multiply a series of transition matrices representing transitions between every two adjacent conceptual levels of the link hierarchy or conceptual link hierarchy. Thus computational costs can be dramatically reduced.

Spears [1998] proposed a *transition matrix compression algorithm*, which compresses a sparse transition matrix to a much smaller size by aggregating states similar in terms of their transition behavior. When raised to the$n$th power, the compressed matrix can still approximate the original matrix well. In order to improve the efficiency of using a MMS for link prediction, we apply the compression algorithm to the MMS.

In using three kinds of Markov chain models for link prediction, we propose to predict the *most probably to-be-visited (MPT)*pages within the next m steps as an improvement of Sarukkai's work [1998], which predicts the MPT pages within the next step. There are two reasons for the improvement. First, by taking into account more steps in the future, prediction results can help users save more clicks and time. Second, prediction results using a MMH or MMC can be mapped onto different conceptual levels of a link hierarchy or conceptual link hierarchy. By taking into account more steps in the future, we can help users explore deeper into the link hierarchy or conceptual link hierarchy to find desired information. In the next step, users are not limited to choose a page linked by the current page, and can directly jump to a page or cluster which contains desired information but is many links away from the current page.

Chen et al. [1998] proposed a*maximal forward path method*that discards backward links in a user sequence. These backward links were made mainly for ease of navigation and thus deemed irrelevant to a user's task during a visit. We apply the method to a user sequence[6] before it is used for link prediction in order to improve the accuracy of link prediction. We further propose to improve the method when it is applied to a user sequence on a link hierarchy or conceptual link hierarchy. Due to the influence of caching, some pages in a user sequence may not be recorded. Using the structural information of a link hierarchy or conceptual link hierarchy, we can accurately recover these missing pages in a user sequence.

Along with using a MMH or MMC for link prediction, we propose *guided paths* to help users navigate deeply into a link hierarchy or conceptual link hierarchy. Given a user's current page on the lth level of a link hierarchy consisting of L levels, we predict the MPT pages from the $(l+1)$th level to the Lth level. A guided path consists of a series of linked MPT pages from on the $(l+1)$th level to the Lth level.

Link prediction using a MMH or MMC is integrated with the visualized link hierarchy or conceptual link hierarchy for adaptive Web site navigation. Link prediction using three kinds of Markov chain models are compared in terms of their effectiveness and efficiency in helping users find desired information on a Web site.

4.2 Overview

As discussed in chapter 2, Zukerman et al. [1999] proposed to use a collaborative approach that uses a Markov chain model regarding the behavior of a group of users to predict an individual user's actions on the WWW. Sarukkai [2000], Dhyani et al. [2002] and many others used a Markov chain model to predict an individual user's navigation on the WWW. Built upon their work, we propose to use three kinds of Markov chain models for adaptive user navigation on a Web site. First, we propose to construct three kinds of Markov chain models from a Web site link structure (MMS), a link hierarchy (MMH), and a conceptual link hierarchy (MMC), respectively. A compression algorithm is used to compress the transition matrix of a MMS. Second, we use three kinds of Markov chain models for link prediction, respectively. We propose to use these Markov chain models to predict the most probably to-be-visited (MPT) pages. We propose to use

[6] Due to user privacy concerns, server side adaptation is more applicable than user side adaptation. User sequences cannot be reliably recorded on the server side due to caching and need to be reconstructed.

a maximal forward path method to improve the accuracy of link prediction. We propose to predict the guided paths using a MMH or MMC. Three kinds of Markov chain models are compared in terms of their effectiveness and efficiency in helping users find desired information on a Web site.

4.3 Constructing Markov Chain Models

In this section, we present our approaches to constructing a Markov chain model from a Web site link structure (MMS), a link hierarchy (MMH), and a conceptual link hierarchy (MMC), respectively.

4.3.1 Constructing Markov Chain Models from Web Site Link Structures (MMSs)

In this section, we present our approaches to constructing a Markov chain model from a Web site link structure (MMS). First, we construct a Markov chain model from the Web site link structure (MMS). Second, we use a matrix compression algorithm to compress the MMS for efficient link prediction.

4.3.1.1 Constructing Markov Chain Models from Web Site Link Structures (MMSs)

After constructing the link hierarchy of a Web site using the method proposed in chapter 3, we add all the secondary links to the link hierarchy and get the Web site link structure. We add the “Start” node to the link structure as the starting point for the user’s visit to the Web site and the “Exit” node as the ending point of the user’s visit. In order to ensure that there is a directed path between any two nodes in the link structure, we add a link from the “Exit” node to the “Start” node.

Figure 4.1 shows a link structure of the University of Ulster Web site. The title of each page is synthesized using the method proposed in chapter 3.

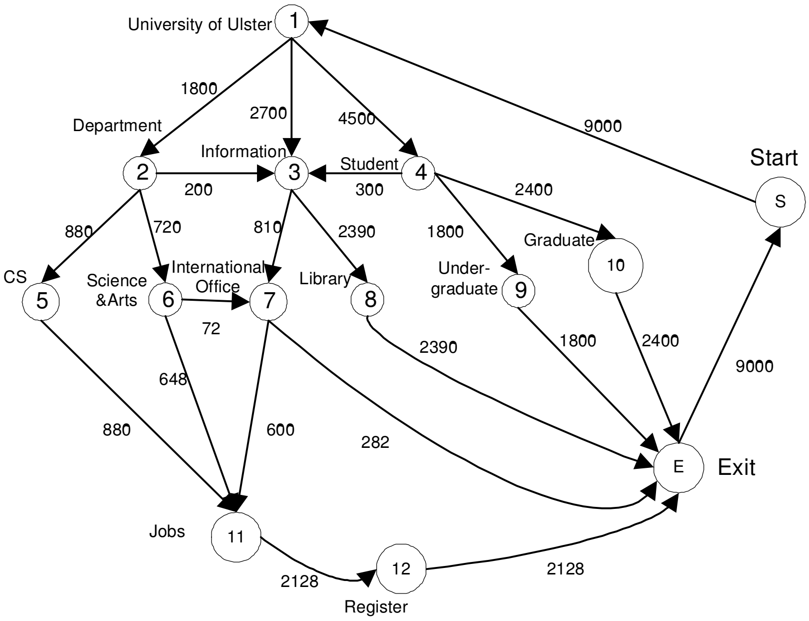


Figure 4.1: A Web site link structure.

A user's navigation process on a Web site link structure can be modeled as a *stochastic process* that consists of a sequence of events $\{X_n\}$, $n = 0, 1, 2, \ldots$, where n identifies the steps in the sequence, and X_n can take any value from a state space consisting of all the pages in the link structure $S = \{1, \ldots, m\}$. The outcome of a future event can be seen to be dependent only on a fixed number of previous events, i.e., we can assume that X_{n+1} is dependent only on i previous events, where $i \geq 1$ is a fixed and finite number. In this case, obtaining the probability of visiting a particular page in the $(n+1)$th step, $P\{X_{n+1} = j\}$, requires only the information about the previous i events, i.e., $P\{X_{n+1} = j \mid X_0 = i_0, \ldots, X_n = i_n\} = P\{X_{n+1} = j \mid X_{n-i+1} = i_{n-i+1}, \ldots, X_n = i_n\}$. We call this sequence of events an *i th-order Markov chain*. When the next event depends on only the current event, i.e., X_{n+1} is only dependent on X_n, the chain is a *first-order Markov chain*[7].

[7] Unless specifically stated, we refer to a first-order Markov chain as a Markov chain.

We use a first-order Markov chain instead of a higher-order Markov chain to model user navigation for two reasons. First, due to the influence of caching, pages visited by users during their visits to a Web site cannot be reliably recorded in Web log files [Pirolli and Pitkow 1999]. Second, there is no reliable method for user identification due to the influence of reasons such as dynamic IP and the same computer shared by multiple users [Pirolli and Pitkow 1999].

The Markov chain of user navigation on a Web site link structure is *ergodic* [Ross 1983] since it is possible to go from every state to every other state in one or more transitions. Since the behavior of a Markov chain at time $n+1$ is dependent only on its behavior at time n, the chain can be described completely by its one-step transition probabilities at step n. The *one-step transition probability* from page i to page j, $P_{i,j}$, can be viewed as the fraction of traversals from i to j over the total number of user traversals from i to the other pages and the "Exit" node.

$$P_{i,j} = P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i) = \frac{w_{i,j}}{\sum_k w_{i,k}} \qquad (4.1)$$

where $w_{i,j}$ is the weight on the link from i to j, and $\sum_k w_{i,k}$ is the sum of weights on all the out-links of page i. The *transition matrix* represents the one-step transition probability between any two states. In the transition matrix, row i contains one-step transition probabilities from i to all states. Row i sums up to 1.0. Column i contains one-step transition probabilities from all states to i. The transition matrix calculated from the link structure in Figure 4.1 is shown in Figure 4.2.

State \ *State*	1	2	3	4	5	6	7	8	9	10	11	12	*Exit*	*Start*
1		0.2	0.3	0.5										
2			0.111		0.489	0.4								
3							0.253	0.747						
4			0.067						0.4	0.533				
5											1.0			
6							0.1				0.9			
7											0.68		0.32	
8													1.0	
9													1.0	
10													1.0	
11												1.0		
12													1.0	
Exit														1.0
Start	1.0													

Figure 4.2: Transition probability matrix of a Markov chain on the link structure in Figure 4.1.

The *initial probability distribution* of pages, $\pi^{(0)}$, can be estimated from training data, e.g., a Web log file.

$$\pi^{(0)}_{(i)} = \frac{w_i}{\sum_k w_k} \tag{4.2}$$

where w_i is the number of appearances of page i in the training data, and $\sum_k w_k$ is the total number of appearances of all the pages in the training data. For a link structure, w_i is the sum of weights on all the in-links of page i, and $\sum_k w_k$ is the sum of weights on all the links in the link structure.

The initial state distribution of the link structure in Figure 4.1 is as follows.

$$\pi^{(0)} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & \mathit{Exit} & \mathit{Start} \\ 0.181 & 0.036 & 0.064 & 0.09 & 0.018 & 0.014 & 0.018 & 0.048 & 0036 & 0.048 & 0.043 & 0.043 & 0.181 & 0.181 \end{pmatrix}$$

A *Markov chain model* constructed from a Web site link structure (MMS) is defined as a three-tuple (S, P, $\pi^{(0)}$), where S is the state space consisting of all the states in a link structure, P is the one-step probability transition matrix. A Markov chain is a sequence of state distribution vectors at successive steps, that is, ($\pi^{(0)}$, $\pi^{(1)}$, …, $\pi^{(n)}$).

4.3.1.2 Compressing MMSs

One-step transition behavior of a MMS consisting of N states is described by an $N \times N$ transition matrix P. The *n-step transition behavior* is described by the nth power of P, i.e., P^n. For a large P this is computationally expensive since the computational complexity of P^n is $O(N^3)$.

Spears [1998] proposed a compression algorithm, which can compress a sparse transition matrix to a smaller size while the transition behavior of the Markov chain model is preserved. Spears' experiments showed that a transition matrix P can be compressed to a smaller matrix P_c without causing significant errors. The time taken by compression can be compensated by all subsequent probability computations for link prediction.

As opposed to other methods for reducing the number of states in a Markov chain model, which have focused on providing good estimations of the steady-state behavior of the Markov chain model [Stewart 1994], Spears' algorithm compresses a transition matrix based on its transition behavior, where states with similar transition behaviors are aggregated together to form new states. Spears [1998] pointed out that if the compression algorithm has worked well then the nth power of the compressed matrix P_c should be nearly identical to compressing the nth power of the original matrix P. The error of compression on the transition behavior of the states is measured by($(\mathrm{P}^n)_c$ - $(\mathrm{P}_c)^n$). Perfect compression has occurred if $(\mathrm{P}^n)_c = (\mathrm{P}_c)^n$.

Spears proved that perfect compression can be obtained in two situations. First is "*row equivalence*" in which the two states i and j have identical rows in P, i.e., $\forall k$, $\mathrm{P}_{i,k} = \mathrm{P}_{j,k}$. Second is "*column equivalence*" in which state i has column entries that are a real multiple q of the column entries for state j, i.e., $\forall k$, $\mathrm{P}_{k,i} = q \cdot \mathrm{P}_{k,j}$. It is generally unlikely that pairs of states will be found that are perfectly row equivalent or column equivalent. The goal then is to find a similarity metric that measures the row and column similarity. For any two states i and j, a similarity metric is defined on their columns and rows in P so that similar states when compressed together should yield less errors when the compressed matrix P_c is raised to its nth power [Spears 1998]. Based on the similarity metric defined by Spears [1998], we can see the *transition similarity* of two states i and j as a product of their *in-link* and *out-link similarities*. Their *in-link similarity* is a weighted distance between column i and column j. Their *out-link similarity* is a distance between row i and row j.

$$\begin{aligned} Sim_{i,j} &= Sim_{i,j}^{out} \times Sim_{i,j}^{in} \\ Sim_{i,j}^{out} &= \sum_y | P_{i,y} - P_{j,y} | \\ Sim_{i,j}^{in} &= \sum_x \left| \frac{m_i \times P_{x,j} - m_j \times P_{x,i}}{m_i + m_j} \right| \\ m_i &= \sum_l P_{l,i} \text{, } m_j = \sum_l P_{l,j} \end{aligned} \qquad (4.3)$$

where m_i and m_j are the sums of the probabilities on the in-links of state i and j respectively, $Sim_{i,j}^{out}$ is the out-link similarity, and $Sim_{i,j}^{in}$ is the in-link similarity.

For the transition matrix in Figure 4.2, the transition similarity matrix is shown in Figure 4.3.

*State**State*	1	2	3	4	5	6	7	8	9	10	11	12	*Exit*	*Start*
1	0.00													
2	0.58	0.00												
3	1.29	0.21	0.00											
4	1.24	0.00	0.36	0.00										
5	1.31	0.57	0.74	0.99	0.00									
6	1.14	0.53	0.60	0.89	0.00	0.00								
7	1.04	0.51	0.81	0.83	0.26	0.24	0.00							
8	1.71	0.63	1.17	1.20	1.18	1.04	0.18	0.00						
9	1.14	0.53	0.75	0.89	0.88	0.80	0.51	0.00	0.00					
10	1.39	0.58	0.87	1.03	1.02	0.91	0.58	0.00	0.00	0.00				
11	2.88	0.74	1.61	1.68	1.64	1.38	0.89	2.32	1.39	1.77	0.00			
12	2.00	0.67	1.29	1.33	1.31	1.14	0.71	0.00	0.00	0.00	2.88	0.00		
Exit	3.25	0.76	1.72	1.79	1.75	1.46	1.31	2.55	146	1.90	5.98	3.25	0.00	
Start	2.00	0.67	1.29	1.33	1.31	1.14	1.04	1.71	1.14	1.39	2.88	2.00	3.25	0.00

Figure 4.3: Transition similarity matrix (symmetric) for the transition matrix in Figure 4.2.

If the similarity is close to zero, the*error* resulted from compression is close to zero [Spears 1998]. We can set a*threshold* ε, and let $Sim_{i,j} < \varepsilon$ to look for candidate states for compression. By raising ε, we can compress more states with an increased error accordingly. By setting an appropriate thresholdε, the error of compression on the transition behavior of the states, i.e., $((\mathrm{P}^n)_c - (\mathrm{P}_c)^n)$ can be controlled, the transition behavior of the states is preserved and the Markov chain model is compressed to an optimal size for efficient link prediction. Intuitively, pages sharing more in-links, out-links, and having equivalent weights on those links are better candidates for compression. Suppose two states i and j are merged together, we need to assign transition probabilities between the new merged state$i \vee j$ and each of the remaining states k in the transition matrix. Spears [1998] used a weighted average of the$i$th and jth rows as the row of state $i \vee j$, and take the sum of the ith and jth columns as the column of state $i \vee j$.

$$\begin{aligned} P_{k,i \vee j} &= P_{k,i} + P_{k,j} \\ P_{i \vee j,k} &= \frac{m_i \times P_{i,k} + m_j \times P_{j,k}}{m_i + m_j} \end{aligned} \qquad (4.4)$$

Spears' experiments indicated that a value ofε between 0.08 and 0.15 yielded good compression, i.e., the size of the transition matrix is effectively reduced and error is small. For the similarity matrix in Figure 4.3, we setε=0.10. The compression process is shown in Figure 4.4.

```
Compressed state 4 into state 2 (similarity 0.000000)(states: 2 4
Compressed state 6 into state 5 (similarity 0.000000)(states: 5 6)
Compressed state 9 into state 8 (similarity 0.000000)(states: 8 9)
Compressed state 12 into state 10 (similarity 0.000000)(states: 10 12)
Compressed state 10 into state 8 (similarity 0.000000)(states: 8 9 10 12)
Finished compression.
Have compressed 14 states to 9.
```

Figure 4.4: Compression process for transition matrix in Figure 4.2.

States 2 and 4, 5 and 6 are compressed since they are column equivalent, i.e., $Sim^{in}_{i,j}$=0. States 8, 9, 10 and 12 are compressed since they are row equivalent, i.e., $Sim^{out}_{i,j}$ =0.

The *compressed transition matrix* containing the new transition probabilities between the new merged states and the other states is shown in Figure 4.5. The compressed matrix is denser than the original transition matrix.

$_{Page}\backslash^{Page}$	1	(2,4)	3	(5,6)	7	(8,9,10,12)	11	*Exit*	*Start*
1		0.7	0.3						
(2,4)			0.08	0.25		0.67			
3					0.25	0.75			
(5,6)					0.04		0.96		
7							0.68	0.32	
(8,9,10,12)								1.0	
11						1.0			
Exit									1.0
Start	1.0								

Figure 4.5: Compressed transition matrix for transition matrix in Figure 4.2.

There is no compression error for the transition matrix in Figure 4.2 since all compressed states are either row or column equivalent. It may not be the case for a transition matrix calculated from another link structure.

Based on the compressed transition matrix in Figure 4.5, we can construct a compressed link structure in Figure 4.6 of the link structure in Figure 4.1. We simply take the sum of the numbers of traversals on the in-links and out-links of pages in a compressed state as the numbers of traversals on the in-links and out-links of the compressed state respectively.

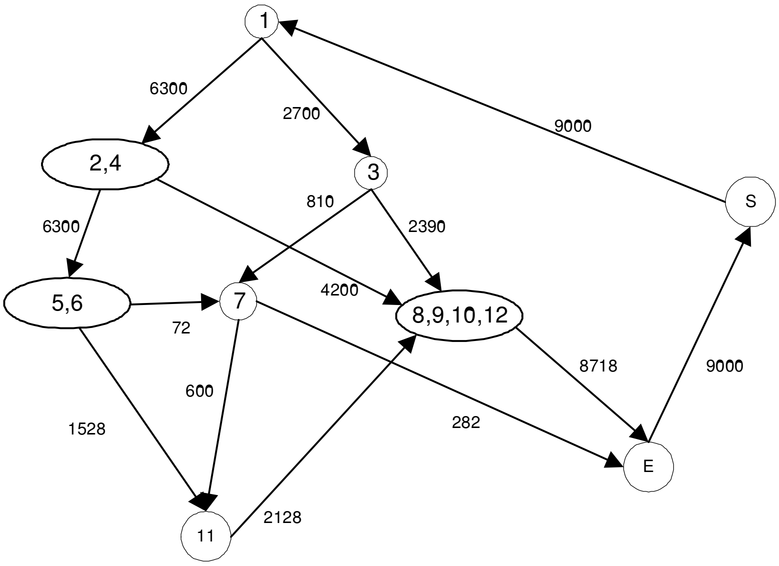


Figure 4.6: Compressed link structure of the link structure in Figure 4.1.

4.3.2 Constructing Markov Chain Models from Link Hierarchies (MMHs) and Conceptual Link Hierarchies (MMCs)

In this section, we present our approaches for constructing a Markov chain model from a link hierarchy (MMH) and a conceptual link hierarchy (MMC) respectively. First, we propose a unified approach for constructing a MMH and MMC from a link hierarchy and a conceptual link hierarchy. Second, we propose a method of computing the n th-power of the transition matrix of a MMH or MMC efficiently.

4.3.2.1 Constructing MMHs and MMCs

Consider a *link hierarchy* or *conceptual link hierarchy* consisting of k conceptual levels, L_1, …, L_k. Each level L_i consists of n_i nodes representing pages and clusters, $P_{i,1}$, $P_{i,2}$, …, P_{i,n_i}. Links are from a level L_i to its adjacent lower level L_{i+1}. We add the "Start" node to the hierarchy[8] as the starting point of users' visits to the Web site and the

[8] A link hierarchy or conceptual link hierarchy.

"Exit" node as the ending point of users' visits. We add a link from the "Start" node to the home page. We add a link from a node in the hierarchy to the "Exit" node if the sum of user traversals on the in-links of the node is larger than the sum of user traversals on the out-links of the node. We assign the difference between the two sums as the weight on the link. We add a link from the "Exit" node to the "Start" node. The Markov chain on the hierarchy is *ergodic*.

A *first-order Markov chain model* is defined as a three-tuple (S, P, $\pi^{(0)}$). S is the state space with each state consisting of a node and its level, i.e., $S = \{(P, L)\}$, where P is a node and L is its level. P is a probability transition matrix representing one-step transition probabilities between states. The one-step transition probability from node $P_{i,k}$ on level i to node $P_{i+1,l}$ on level $i+1$, $\mathrm{P}_{(i,k),(i+1,l)}$, can be viewed as the fraction of traversals from $P_{i,k}$ to $P_{i+1,l}$ over the total number of traversals from $P_{i,k}$ to all the nodes on level $i+1$ and the "Exit" node.

$$\mathrm{P}_{(i,k),(i+1,l)} = \frac{w_{(i,k),(i+1,l)}}{w_{(i,k),"Exit"} + \sum_j w_{(i,k),(i+1,j)}} \tag{4.5}$$

where $w_{(i,k),(i+1,l)}$ is the weight on the link from $P_{i,k}$ to $P_{i+1,l}$, $\sum_j w_{(i,k),(i+1,j)}$ is the sum of weights on all the out-links of $P_{i,k}$, and $w_{(i,k),"Exit"}$ is the weight on the link from $P_{i,k}$ to the "Exit" node. The *transition matrix* P represents the transition probabilities between nodes on every two adjacent conceptual levels of the hierarchy. In the transition matrix, row i contains transition probabilities from i to the nodes on the adjacent lower conceptual level and the "Exit" node. Row i sums up to 1.0. Column i contains transition probabilities to i from the nodes on the adjacent higher conceptual level and the "Start" node.

A link hierarchy of the link structure in Figure 4.1 with the "Exit" and "Start" nodes is shown in Figure 4.7.

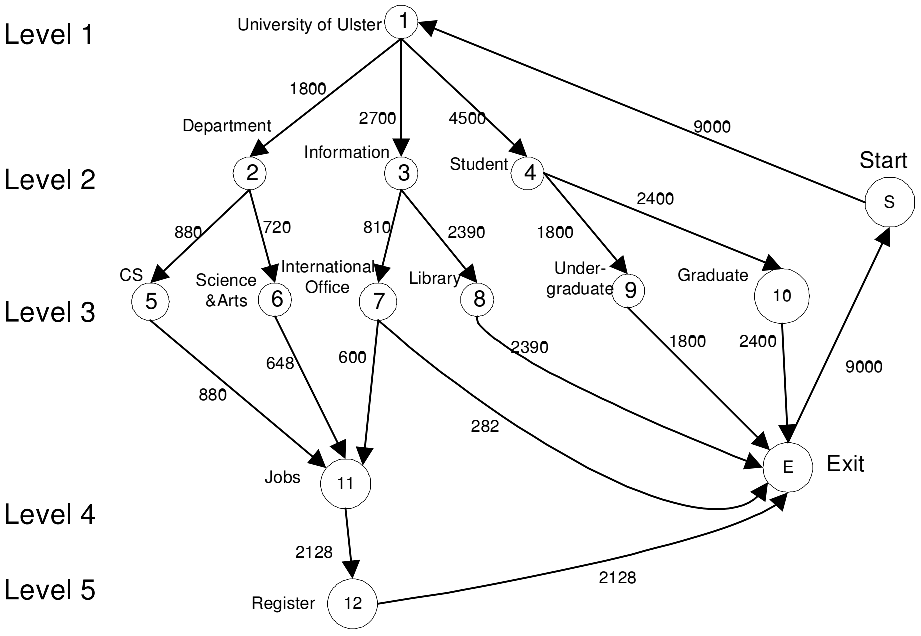


Figure 4.7: A link hierarchy with "Exit" and "Start" nodes.

The transition matrix of the link hierarchy is shown in Figure 4.8.

Page \ *Page*	1,1	2,2	3,2	4,2	5,3	6,3	7,3	8,3	9,3	10,3	11,4	12,5	*Exit*	*Start*
1,1		0.2	0.3	0.5										
2,2					0.55	0.45								
3,2							0.253	0.747						
4,2									0.429	0.571				
5,3											1.0			
6,3											1.0			
7,3											0.68		0.32	
8,3													1.0	
9,3													1.0	
10,3													1.0	
11,4												1.0		
12,5													1.0	
Exit														1.0
Start	1.0													

Figure 4.8: Transition matrix of the link hierarchy in Figure 4.7.

The *initial state distribution*, $\pi^{(0)}$, calculated from the link hierarchy in Figure 4.7 is as follows.

$$\pi^{(0)} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & \mathit{Exit} & \mathit{Start} \\ 0.183 & 0.036 & 0.055 & 0.091 & 0.018 & 0.015 & 0.016 & 0.049 & 0.037 & 0.049 & 0.043 & 0.043 & 0.183 & 0.183 \end{pmatrix}$$

A conceptual link hierarchy with “Exit” and “Start” nodes constructed from the link hierarchy in Figure 4.7 using the PageCluster algorithm proposed in chapter 3 is shown in Figure 4.9.

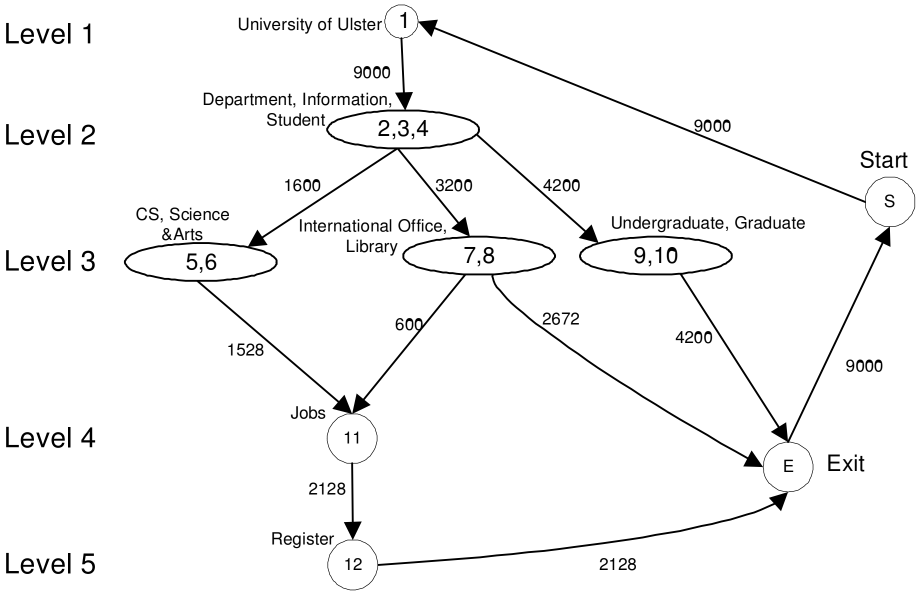


Figure 4.9: A conceptual link hierarchy with “Exit” and “Start” nodes constructed from the link hierarchy in Figure 4.7.

The transition matrix of the conceptual link hierarchy in Figure 4.9 is shown in Figure 4.10.

Node\Node	1,1	(2,3,4),2	(5,6),3	(7,8),3	(9,10),3	11,4	12,5	*Exit*	*Start*
1,1		1.0							
(2,3,4),2			0.178	0.356	0.467				
(5,6),3						1.0			
(7,8),3						0.183		0.817	
(9,10),3								1.0	
11,4							1.0		
12,5								1.0	
Exit									1.0
Start	1.0								

Figure 4.10. Transition matrix of the conceptual link hierarchy in Figure 4.9.

The initial probability distribution of states, $\pi^{(0)}$, calculated from the conceptual link hierarchy in Figure 4.9 is as follows.

$$\pi^{(0)} = \left(\begin{array}{ccccccccc} 1 & 2,3,4 & 5,6 & 7,8 & 9,10 & 11 & 12 & Exit & Start \\ 0.183 & 0.183 & 0.032 & 0.065 & 0.085 & 0.043 & 0.043 & 0.183 & 0.183 \end{array}\right)$$

4.3.2.2 A Method for Computing the n th-Power of Transition Matrices of MMHs and MMCs

If the dimension of P is large, computation of P^k can be expensive. By taking into account conceptual levels in a link hierarchy or conceptual link hierarchy, the computation cost of P^k can be dramatically reduced.

Links from nodes on a hierarchy to the "Exit" node may be across multiple conceptual levels. We add *dummy nodes* representing the "Exit" node to conceptual levels of a hierarchy and transform these links into structural links. The link hierarchy in Figure 4.7 with dummy nodes is in Figure 4.11.

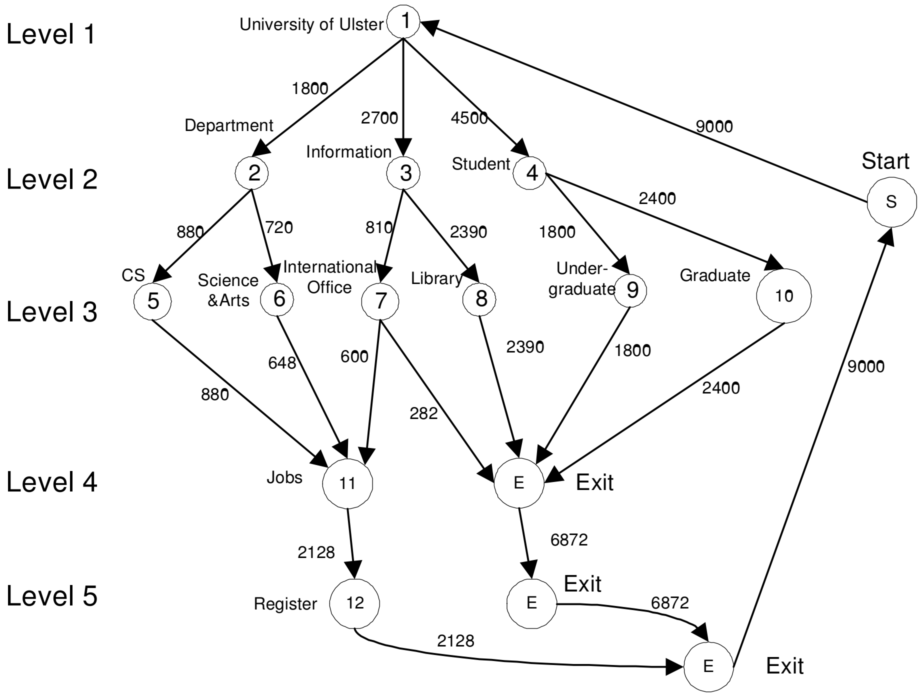


Figure 4.11: A link hierarchy with two dummy "Exit" nodes.

The transition matrix P of a hierarchy, which has dummy "Exit" nodes and n conceptual levels, is:

$$
\mathrm{P}=\begin{bmatrix}
0 & \mathrm{P}_{L_1,L_2} & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \mathrm{P}_{L_2,L_3} & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \ddots & \ddots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \ddots & \mathrm{P}_{L_{n-2},L_{n-1}} & 0 & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \mathrm{P}_{L_{n-1},L_n} & 0 & 0 \\
0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \mathrm{P}_{L_n,Exit} & 0 \\
0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \mathrm{P}_{Exit,Start} \\
\mathrm{P}_{Start,L_1} & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0
\end{bmatrix}
$$

where $\mathrm{P}_{L_i,L_{i+1}}$ ($i=1$, ..., n-1) represents transition probabilities between the nodes on the ith level and the nodes on the $(i+1)$th level of the hierarchy, $\mathrm{P}_{L_n,Exit}$ represents transition probabilities between the nodes on the nth level and the "Exit" node, $\mathrm{P}_{Exit,Start}$ represents the transition probability between the "Exit" and "Start" node, and P_{Start,L_1} represents the transition probability between the "Start" and the home page node. We set $\mathrm{P}_{Exit,Start}=\mathrm{P}_{Start,L_1}=1$.

We can compute P^2, P^3, and P^k as follows.

$$\mathrm{P}^2 =$$

$$\begin{bmatrix}
0 & 0 & \mathrm{P}_{L_1,L_2}\mathrm{P}_{L_2,L_3} & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & \mathrm{P}_{L_2,L_3}\mathrm{P}_{L_3,L_4} & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathrm{P}_{L_3,L_4}\mathrm{P}_{L_4,L_5} & \ddots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \ddots & 0 & \vdots & \vdots \\
\vdots & \vdots & \vdots & 0 & 0 & \ddots & \mathrm{P}_{L_{n-2},L_{n-1}}\mathrm{P}_{L_{n-1},L_n} & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & \mathrm{P}_{L_{n-1},L_n}\mathrm{P}_{L_n,L_{Exit}} & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathrm{P}_{L_n,L_{Exit}} \\
1 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 \\
0 & \mathrm{P}_{L_1,L_2} & 0 & \cdots & \cdots & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$
\mathrm{P}^3 =
\left[\begin{array}{ccc:cccccc}
0 & 0 & 0 & \mathrm{P}_{L_1,L_2}\mathrm{P}_{L_2,L_3}\mathrm{P}_{L_3,L_4} & 0 & & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathrm{P}_{L_2,L_3}\mathrm{P}_{L_3,L_4}\mathrm{P}_{L_4,L_5} & & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \cdot^{\cdot^{\cdot}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \cdot^{\cdot^{\cdot}} & \mathrm{P}_{L_{n-1},L_{n-2}}\mathrm{P}_{L_{n-2},L_{n-1}}\mathrm{P}_{L_{n-1},L_n} & & \\
 & & & 0 & 0 & \cdot^{\cdot^{\cdot}} & 0 & \mathrm{P}_{L_{n-2},L_{n-1}}\mathrm{P}_{L_{n-1},L_n}\mathrm{P}_{L_n,L_{Exit}} & \\
 & & & & & \cdot^{\cdot^{\cdot}} & 0 & 0 & \mathrm{P}_{L_{n-1},L_n}\mathrm{P}_{L_n,L_{Exit}} \\
\hdashline
\mathrm{P}_{L_n,L_{Exit}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathrm{P}_{L_1,L_2} & 0 & & & 0 & 0 & 0 & 0 \\
0 & 0 & \mathrm{P}_{L_1,L_2}\mathrm{P}_{L_2,L_3} & & & 0 & 0 & 0 & 0
\end{array}\right]
$$

$P^k =$

$$\left[\begin{array}{cccc|ccccc}
0 & 0 & \cdots & 0 & \prod_{i=1}^{k} P_{L_i,L_{i+1}} & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & \ddots & 0 & \vdots & 0 \\
\vdots & 0 & 0 & 0 & \vdots & 0 & \prod_{i=1}^{k} P_{L_{n-i},L_{n-i+1}} & 0 & \vdots \\
\vdots & \vdots & 0 & 0 & 0 & \ddots & 0 & P_{L_n,L_{Exit}} \prod_{i=1}^{k-1} P_{L_{n-i},L_{n-i+1}} & 0 \\
0 & 0 & \vdots & 0 & 0 & \ddots & \vdots & 0 & P_{L_n,L_{Exit}} \prod_{i=1}^{k-2} P_{L_{n-i},L_{n-i+1}} \\
\hdashline
P_{L_n,L_{Exit}} & 0 & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & 0 \\
0 & \ddots & 0 & \vdots & \vdots & \cdots & \cdots & 0 & \vdots \\
\vdots & 0 & \ddots & 0 & 0 & \cdots & \cdots & 0 & 0 \\
0 & \cdots & 0 & \prod_{i=1}^{k-1} P_{L_i,L_{i+1}} & 0 & \cdots & \cdots & 0 & 0
\end{array}\right]$$

We can represent P^k as:

$$P^k = \begin{bmatrix} 0 & R \\ Q & 0 \end{bmatrix}$$

where R is a matrix whose diagonal consists of $\prod_{i=1}^{k} P_{L_i,L_{i+1}}$, …, $\prod_{i=1}^{k} P_{L_{n-i},L_{n-i+1}}$, $P_{L_n,L_{Exit}} \prod_{i=1}^{k-1} P_{L_{n-i},L_{n-i+1}}$, $P_{L_n,L_{Exit}} \prod_{i=1}^{k-2} P_{L_{n-i},L_{n-i+1}}$, and Q is a matrix whose diagonal consists of $P_{L_n,L_{Exit}}$, ..., $\prod_{i=1}^{k-1} P_{L_i,L_{i+1}}$.

We can compute $\prod_{i=1}^{2} P_{L_i,L_{i+1}}$, ..., $\prod_{i=1}^{k} P_{L_i,L_{i+1}}$, …, $\prod_{i=1}^{k} P_{L_{n-i},L_{n-i+1}}$, $P_{L_n,L_{Exit}} \prod_{i=1}^{k-1} P_{L_{n-i},L_{n-i+1}}$, and $P_{L_n,L_{Exit}} \prod_{i=1}^{k-2} P_{L_{n-i},L_{n-i+1}}$ to get P^k. Since the size of $P_{L_i,L_{i+1}}$ is much smaller than the size of P, the computational cost can be dramatically reduced. Instead of storing P^k for link prediction, we store $P_{L_n,L_{Exit}}$, P_{L_1,L_2}, …, $\prod_{i=1}^{2} P_{L_i,L_{i+1}}$, $\prod_{i=1}^{k} P_{L_i,L_{i+1}}$, $\prod_{i=1}^{k} P_{L_{i+1},L_{i+2}}$, …,

$\prod_{i=1}^{k} \mathrm{P}_{L_{n-i},L_{n-i+1}}$, $\mathrm{P}_{L_n,L_{Exit}} \prod_{i=1}^{k-1} \mathrm{P}_{L_{n-i},L_{n-i+1}}$, $\mathrm{P}_{L_n,L_{Exit}} \prod_{i=1}^{k-2} \mathrm{P}_{L_{n-i},L_{n-i+1}}$ and their positions in P^k for link prediction.

4.4 Link Prediction Using Markov Chain Models

In this section, we present our approaches to link prediction using three kinds of Markov chain models respectively. First, we use Markov chain models constructed from Web site link structures (MMSs) for link prediction. Second, we use Markov chain models constructed from link hierarchies (MMHs) and conceptual link hierarchies (MMCs) for link prediction respectively.

4.4.1 Link Prediction Using MMSs

In this section, we propose our approaches to link prediction using MMSs. First, we predict the most probably to-be-visited (MPT) pages given a user sequence. Second, we apply a maximal forward path method to a user sequence in order to improve the accuracy of link prediction.

4.4.1.1 Predicting MPT Nodes Using MMSs

Given a user sequence consisting of n steps, each of which is a Web page, we represent each step i as a *vector* L_{-i+1} with a probability 1.0 if the position represents the page at that step and 0.0 otherwise. We get a list of vectors, L_{-n+1},..., L_{-i+1}, ..., L_0. When L_0 is multiplied by P, we get the probabilities of moving to the other pages in the next step.

$$S_1 = L_0 \cdot \mathrm{P} \qquad (4.6)$$

Sarukkai [2000] proposed a variant of the Markov chain to accommodate weighting of more than one step in a user sequence. For the current step L_0, we only need to go one step further in predicting the next step,S_1. For the last step, L_{-1}, we need to go two steps further in predicting the next step,S_1. For the last (k -1)th step, L_{-k+1}, we need to go k steps further in predicting the next step,S_1. We combine the predictions made by each step in a user sequence as the overall prediction for the next stepS_1.

$$S_1 = a_1 \cdot L_0 \cdot \mathrm{P} + a_2 \cdot L_{-1} \cdot \mathrm{P}^2 + \ldots + a_k \cdot L_{-k+1} \cdot \mathrm{P}^k \qquad (4.7)$$

where we specify $1 > a_1 > \ldots > a_k > 0$ so that the more recent a step is in a user sequence, the more importance it has in predicting the next step.S_1 is then normalized and the pages with the highest probabilities inS_1 are selected as the*most probably to-be-visited (MPT) pages in the next step*

$$Rec_1 = Max(normalized(S_1)) \qquad (4.8)$$

Given the current step, L_0, the *expected number of visits* [Minh 2000], $V_{L_0}^m$, to the other states within the next$m$ steps is as follows.

$$V_{L_0}^m = L_0 \cdot \mathrm{P} + L_0 \cdot \mathrm{P}^2 + \ldots + L_0 \cdot \mathrm{P}^m \qquad (4.9)$$

Given a user sequence, we propose to predict the*most probably to-be-visited (MPT) pages within the next m steps*. The prediction combines the prediction of the MPT pages within the next one step proposed by Sarukkai [2000] and computation of the expected number of visits to the other pages within the next$m$ steps [Minh 2000].

Given a user sequence, which consists of$n$ steps and is represented as a list of vectors $L_{-n+1}, \ldots,\ L_0$, we predict the*MPT pages within the next$m$ steps* as follows.

$$\begin{aligned} S_m = {} & a_{1,1}\times L_0\times \mathrm{P} + a_{1,2}\times L_0\times \mathrm{P}^2 + \ldots + a_{1,m}\times L_0\times \mathrm{P}^n + \\ & a_{2,1}\times L_{-1}\times \mathrm{P}^2 + a_{2,2}\times L_{-1}\times \mathrm{P}^3 + \ldots + a_{2,m}\times L_{-1}\times \mathrm{P}^{n+1} + \\ & \ldots + a_{n,1}\times L_{-n+1}\times \mathrm{P}^n + a_{n,2}\times L_{-n+1}\times \mathrm{P}^{n+1} + \ldots + a_{n,m}\times L_{-n+1}\times \mathrm{P}^{n+m-1} \end{aligned} \tag{4.10}$$

where given a vector L_{-j+1} (j =1, 2, …, n), we specify $1 > a_{j,1} > a_{j,2} > \ldots > a_{j,m} > 0$ so that the nearer a step is to the present vector, L_0, the more importance it has in prediction. In predicting the l th step (l =1, 2, …, m) in the future, we specify $1 > a_{1,l} > a_{2,l} > \ldots > a_{m,l} > 0$ so that the more recent a vector is in a user sequence, the more importance it has in predicting the l th step. S_m is then normalized and the pages with the highest probabilities in S_m are selected as the *most probably to-be-visited (MPT) pages within the next m steps.*

4.4.1.2 Maximal Forward Path Method

Chen et al. [1998] proposed a *maximal forward path* as a sequence of maximally connected pages in a user sequence. Pages revisited mainly for ease of navigation are discarded. These discarded pages are deemed less irrelevant to a user's aim during a visit.

For example, a user followed a page sequence (1, 2, 5, 2, 3) on the link structure in Figure 4.1. Since the user has visited page 5 after page 2 and then gone back to page 2 in order to go to page 6, the maximal forward sequence is (1, 2, 3). We calculate the MPT pages within the next 3 steps, i.e., n =3 and m =3.

$$\begin{aligned} L_0 &= \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \\ L_{-1} &= \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \\ L_{-2} &= \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \end{aligned}$$

Given the transition matrix in Figure 4.2, we set:

$$A = \begin{vmatrix} 1.0 & 0.67 & 0.33 \\ 0.67 & 0.45 & 0.22 \\ 0.33 & 0.22 & 0.11 \end{vmatrix}$$

According to Equation 4.10, we get:

$$S_3 = \begin{vmatrix} 0 & 0 & 0.018 & 0 & 0.032 & 0.026 & 0.329 & 0.886 & 0.066 & 0.088 & 0.756 & 0.473 & 0.951 & 0.375 \end{vmatrix}$$

$$\mathrm{Re}\, c_3 = (8, 11, 12, 7)$$

Pages 8, 11, 12, and 7 are recommended to the user.

4.4.2 Link Prediction Using MMHs and MMCs

In this section, we propose our approaches for link prediction using MMHs and MMCs respectively. First, we use a heuristic approach to improve the maximal forward path method and apply it to a user sequence for more accurate prediction. Second, given a user sequence, we predict MPT pages and clusters and guided paths using MMHs and MMCs respectively.

4.4.2.1 An Improved Maximal Forward Path Method

The influence of *caching* to user sequences has not been taken into account in the maximal forward path method [Chen et al. 1998]. The *improved maximal forward path* method uses the conceptual levels and links of a link hierarchy to reconstruct user sequences distorted by caching.

Consider a user sequence consisting of m pages, (P_1, …, P_i, P_{i+1}, …, P_m) on a link hierarchy. Suppose there is no link (either a structural link or a secondary link) from P_i to P_{i+1}. We try to infer the most probable *missing pages* between P_i and P_{i+1} to form a path between them. Three cases need to be considered in terms of the conceptual levels of P_i and P_{i+1}. First, P_{i+1} is on a lower level than P_i. Second, P_{i+1} is on the same level as P_i. Third, P_{i+1} is on a higher level than P_i.

Both *server level* and *user level caching* may have affected a user sequence. *Proxy servers* may cache documents that were previously requested by a group of users and send these documents to an individual user who request them for the first time. An individual user's computer only caches documents that were previously requested by the user.

Our heuristic approach is as follows. *Sever level caching* has influence in the first case, i.e., some pages between P_{i+1} and P_i have been cached on the proxy server. *User level caching* has influence in the second and third cases, i.e., some pages between P_{i+1} and P_i have been cached on the user side computer. *User level caching* may also have influence in the first case, and *server level caching* may also have influence in the second and third cases.

If only *user level caching* was present, we can find a page between P_1 and P_{i-1} which links to P_{i+1}. If we cannot find a page between P_1 and P_{i-1} which links to P_{i+1}, *server level caching* was present. We instead use the link hierarchy to infer a path from one page between P_1 and P_{i-1} to P_{i+1}.

The heuristic approach takes a user sequence consisting of m pages, (P_1, …, P_i, P_{i+1}, …, P_m), as it input. For each page P_i $i=2, 3, \ldots, m$ in the sequence, if we can find page P_k ($k=i$, …, 1) which links to P_{i+1}, we remove pages P_{k+1}, …, P_i from the sequence, and return (P_1, …, P_k, P_{i+1}) as the *prefix* of the user sequence. Otherwise, server level caching was present and we infer a path from P_k ($k=i$, …, 1) to P_{i+1}. If page P_k ($k=i$, …, 1) is on a higher level than P_{i+1}, server level caching was present. If using a *breadth-first search method* we can find a path from P_k to P_{i+1} as (P_k, …, P_{i+1}), (P_1, …, P_k, …, P_{i+1}) is returned as the prefix of the user sequence. Otherwise, we go back to the main parent of P_1, P_{r1}, and use a *breadth-first search method* to find a path from P_{r1} to P_{i+1}. If we can find the path, (P_{r1}, …, P_{i+1}) is returned as the *prefix* of the user sequence. Otherwise, we go back to the main parent of P_{r1}, P_{r2}, and use a *breadth-first search method* to find a path from P_{r2} to P_{i+1}. If we can find the path, we return (P_{r2}, …, P_{i+1}) as the *prefix* of the user sequence. This process continues until we find a path to be returned as the*prefix* of the user sequence. In the worst cases, we go back to the home page, P_H, which is the root of the link hierarchy, and use a*breadth-first search method* to find a path to P_{i+1} and (P_H, ..., P_{i+1}) is returned as the*prefix* of the user sequence.

For example, consider a user sequence (1, 2, 8, 12) on the link hierarchy with secondary links in Figure 4.12.

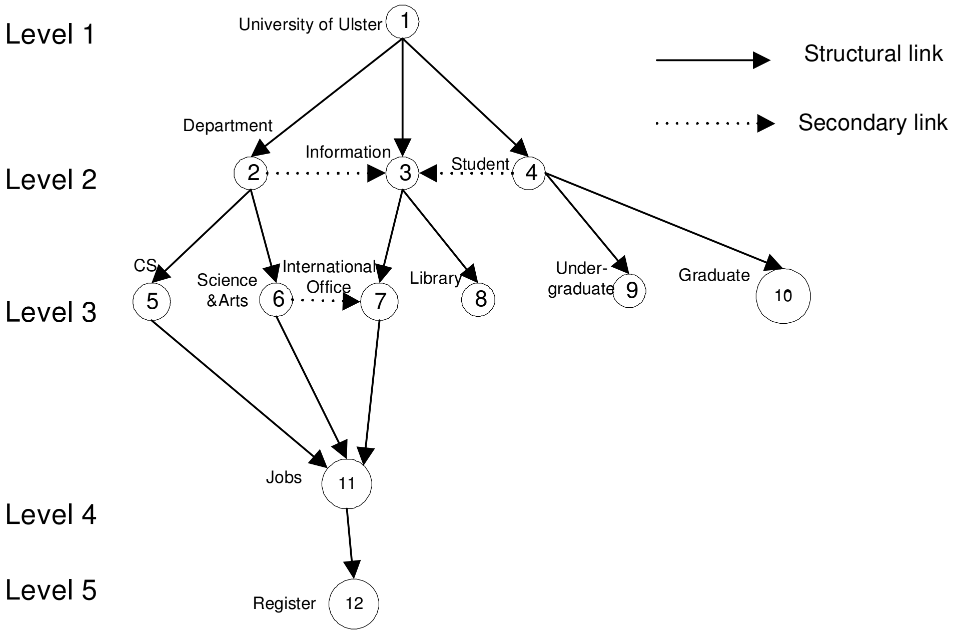


Figure 4.12: The link hierarchy in Figure 4.7 with secondary links.

Starting from page 2, we get the prefix (1, 2). For page 8, we cannot find a path from page 2 to page 8, but we find a path from page 1 to 8 as (1, 3, 8). For page 12, we cannot find a path from 8 to 12, but we find a path from page 3 to 12 as (3, 7, 11, 12). So the maximal forward sequence is (1, 3, 7, 11, 12). We can convert the sequence into a maximal forward sequence of pages and clusters on the conceptual link hierarchy in Figure 4.9 as (1, (2, 3, 4), (7, 8), 11, 12).

Consider another user sequence (2, 8, 12). Starting from 8, we cannot find a path from page 2 to 8. So we go back to main parent of page 2, which is page 1. We find a path from page 1 to 8 as (1, 3, 8). For page 12, we find a path from page 3 to 12 as (3, 7, 11, 12). So the maximal forward sequence is (1, 3, 7, 11, 12) as well.

4.4.2.2 Predicting MPT Nodes and Guided Paths Using MMHs and MMCs

Given a maximal forward sequence, we can use Equation 4.10 to predict MPT pages and clusters within the next m steps on a link hierarchy or conceptual link hierarchy respectively.

We propose to predict *guided paths* given a user's current node i on a certain conceptual level l of a link hierarchy or conceptual link hierarchy. A *guided path* of length m consists of a linked sequence of m nodes on the $(l+1)$th to $(l+m)$th levels respectively.

The current node i is represented as a vector L_0. The probability distributions of the next step, second step to mth step are $S_1 = L_0 \cdot \mathrm{P}$, $S_2 = L_0 \cdot \mathrm{P}^2$, …, $S_m = L_0 \cdot \mathrm{P}^m$, respectively. S_1, S_2, …, S_m are normalized. The pages with the highest probabilities in S_1, S_2, …, S_m are selected as the *most probably to-be-visited (MPT) pages* at the first, second, …, mth step, respectively. MPT pages from each step are concatenated to form *guided paths.*

For example, for the link hierarchy in Figure 4.7 and its transition matrix in Figure 4.8, given a maximal forward sequence of a user as (1, 2), we can predict MPT pages within the next two steps. We set $a_{1,1}$=1.0, $a_{1,2}$=0.5, $a_{2,1}$=0.5, and $a_{2,2}$=0.25.

$$L_0 = \left| 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right|$$

$$L_{-1} = \left| 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right|$$

Using Equation 4.10, we get:

$$S_2 = \left| 0 \quad 0 \quad 0 \quad 0 \quad 0.605 \quad 0.505 \quad 0.538 \quad 0.112 \quad 0.107 \quad 0.143 \quad 0.069 \quad 0 \; 0.2 \quad 0 \right|$$

Pages 5, 7, 6, and 10 are recommended.

Guided paths can be predicted as follows.

$$S_1 = L_0 \cdot \mathrm{P} = \left| 0 \quad 0 \quad 0 \quad 0 \quad 0.55 \quad 0.45 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right|$$

$$S_2 = L_0 \cdot \mathrm{P}^2 = \left| 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.0 \quad 0 \quad 0 \quad 0 \right|$$

$$S_3 = L_0 \cdot \mathrm{P}^3 = \left| 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.0 \quad 0 \quad 0 \right|$$

MPT pages in the next step are page 5 and 6. MPT pages in the second step are page 11. MPT pages in the third step are page 12. The guided paths are $5 \rightarrow 11 \rightarrow 12$ and $6 \rightarrow 11 \rightarrow 12$.

For the conceptual link hierarchy in Figure 4.9 and its transition matrix in Figure 4.10, given a maximal forward sequence of a user as (1, (2, 3, 4)), we can predict MPT pages and clusters within the next two steps. We set$a_{1,1}$=1.0, $a_{1,2}$=0.5, $a_{2,1}$=0.5, and $a_{2,2}$=0.25.

$$L_0 = \left| 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right|$$
$$L_{-1} = \left| 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right|$$

Using Equation 4.10, we get:

$$S_2 = \left| 0 \quad 0 \quad 0.267 \quad 0.534 \quad 0.7005 \quad 0.153 \quad 0 \quad 0.568 \quad 0 \right|$$

Clusters (9,10), (7,8), and (5,6) are recommended.

Guided paths can be predicted as follows.

$$S_1 = L_0 \cdot \mathrm{P} = \left| 0 \quad 0 \quad 0.178 \quad 0.356 \quad 0.467 \quad 0 \quad 0 \quad 0 \quad 0 \right|$$

$$S_2 = L_0 \cdot \mathrm{P}^2 = \left| 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.243 \quad 0 \quad 0.758 \quad 0 \right|$$

$$S_3 = L_0 \cdot \mathrm{P}^3 = \left| 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.243 \quad 0 \quad 0.758 \right|$$

MPT nodes in the next step are clusters (9,10), (7,8) and (5,6). MPT nodes in the second step are page 11. MPT nodes in the third step are page 12. Guided paths are $(7,8) \rightarrow 11 \rightarrow 12$, and $(5, 6) \rightarrow 11 \rightarrow 12$.

4.5 Comparisons with Related Work

In this section, we compare our approaches with related work. First, we compare first-order Markov chain models with higher-order Markov chain models in link prediction. Second, we compare our work with Sarukkai's work on link prediction. Third, we compare the similarity for compressing a transition matrix with link similarity in the PageCluster algorithm.

4.5.1 Comparing First-Order Markov Chain Models with Higher-Order Markov Chain Models in Link Prediction

In building higher-order Markov chain models, user sequences need to be reconstructed from Web log files. User sequences cannot be reliably reconstructed due to two reasons. First, due to the influence of caching, some pages in user sequences were not recorded in Web log files. Second, there is no reliable method to identify individual users due to dynamic IP allocations and sharing the same computers by multiple users etc. We use referrer information contained in Web log files to construct first-order Markov chain models. Caching can influence the number of user traversals on hyperlinks. Since generally caching cannot distort the overall picture of user requests of documents on a Web site, we can use user traversals to infer one-step transition probabilities between states accurately. User identification is not required in our method. Although higher-order Markov chain models are generally more accurate than first-order Markov chain models in link prediction, a study by Pirolli and Pitkow [1999] suggested that first-order Markov chain models have comparable accuracy in link prediction with higher-order Markov chain models.

4.5.2 Comparing Our Work with Sarukkai's Work on Link Prediction

There are two differences between our work and Sarukkai's work on link prediction. First, we propose to predict the most probably to-be-visited (MPT) pages and clusters within the next m steps given a user sequence. In Sarukkai's work,m is restricted to 1 only. By increasing m, link prediction can provide more insight into the future and help users navigate more deeply into a hierarchy or conceptual link hierarchy. Second, we propose to predict guided paths on a link hierarchy or conceptual link hierarchy. We predict MPT nodes at each step given the current node. MPT nodes at each step are concatenated to form guided paths. Only one node is selected at each step to form a guided tour [Sarukkai 2000]. The node is the MPT node in one step given the last step. If there are more than one MPT node, the node having the maximal URL path prefix matching with the previous step is selected.

4.5.3 Comparing Similarities for Compression with Link Similarities

Both compression similarity and link similarity are defined on the in-links and out-links of a pair of pages or clusters. There are two differences between them. First, link similarity is defined between a pair of pages or clusters on the same conceptual level. We use in-link and out-link similarities separately in clustering. Second, when aggregating two states into a new state in compressing a transition matrix, we compute the compression similarities between the new state and the other state using the new transition matrix. When clustering two clusters into a new cluster in the PageCluster algorithm, link similarity between the new cluster and the other page or cluster is decided by the linkage method in clustering, e.g., complete linkage.

4.6 Summary

In this chapter, we presented our work on using Markov chain models for link prediction toward adaptive Web site navigation. First, we proposed to construct three kinds of Markov chain models from Web site link structures (MMSs), link hierarchies (MMHs), and conceptual link hierarchies (MMCs), respectively. We proposed a method to construct Web site link structures from Web log files. A compression algorithm is used to compress the transition matrix of a MMS for efficient link prediction. We presented a method for computing the nth power of a transition matrix of a MMH or MMC efficiently for link prediction. Second, three kinds of Markov chain models are used for link prediction, respectively. Using all three kinds of models, we proposed to predict

MPT pages within the next m steps given a user sequence. In MMSs, we use a maximal forward path method to improve accuracy of link prediction. In MMHs and MMCs, we improve the maximal forward path method and use it to improve accuracy of link prediction. We proposed to predict guided paths using MMHs and MMCs. Link prediction results using MMHs and MMCs are integrated with link hierarchies and conceptual link hierarchies visualized in ONE for user navigation.

User behavior in the form of traversals on hyperlinks is used to estimate transition probabilities between states of Markov chain models. If user behavior changes, link prediction is influenced in three aspects. First, the compression algorithm compresses the transition matrix differently. Second, a different link hierarchy and conceptual link hierarchy are constructed. Third, transition probabilities between states change. If we use Web log files to record user behavior changes over time to construct multiple Markov chain models, different link prediction results may be generated using these Markov chain models given a user sequence. In our future work, we intend to study the influence of user behavior changes on link prediction using Markov chain models. We also plan to apply our approach to other Web sites.

Chapter 5

ADAPTIVE WEB SITE SEARCH

In this chapter, we present our approaches to ranking search results in response to users' keyword-based queries for adaptive Web site search. We propose the *PageRate* algorithm, which takes into account information about both hyperlinks and user behavior, to give an authority-based ranking, called *PageRate*, to each page. *Extended anchor texts* of a page as other page authors' collective opinions of the page are used to extract for a feature vector representing the page. A *relevance-based ranking* is defined on the two feature vectors representing a page and a user query respectively. The authority-based and relevance-based rankings of a page are integrated as the *overall ranking* of the page in the search results. The chapter is organized as follows. In section 5.1, we discuss the problems this chapter addresses. In section 5.2, we give an overview of our approaches. In section 5.3, we present the PageRate algorithm to give authority-based rankings to pages. In section 5.4, we combine PageRates with relevance-based rankings of pages to rank search results. In section 5.5, we compare our approaches with related work. Finally we conclude in section 5.6.

5.1 Motivation

Navigation and search are the two predominant paradigms for finding information on the Web [Olston and Chi 2003]. By combining navigation and search, users may be able to find desired information more effectively and efficiently.

Usually a user uses a keyword-based query to search for information on a Web site. The content of each page is compared with the query to get a relevance-based ranking of the page. Pages can be ordered in terms of their relevance-based rankings in search results. However, relevance-based rankings of pages cannot guarantee the returned pages are *authoritative* on the query topic. Most queries consist of one to three keywords. For a Web site having a large number of pages, searches typically return hundreds of pages, among which there are only a few authoritative ones. Relevance-based rankings are also susceptible to *keyword spamming* [Chakrabarti 2002].

On the other hand, the *PageRank* algorithm [Page et al. 1998] uses information about hyperlinks to give an authority-based ranking, called *PageRank*, to each page. In the PageRank algorithm, a link from one page A to another page B is seen as a *recommendation* of page B by the author of page A. The PageRank of a page is given by the PageRanks of the pages which link to it. Their PageRanks again are given by the PageRanks of pages which link to them. Hence, PageRank of a page is always determined recursively by the PageRanks of other pages. Combining relevance-based and authority-based rankings of pages in ranking search results, users can search for pages both *relevant* and *authoritative* to their queries effectively and efficiently.

We propose to incorporate user behavior in ranking pages. We assume that the opinion of an individual user in following a hyperlink can be characterized by the collective opinions of a group of users in following the hyperlink. The PageRate algorithm as an improvement of the PageRank algorithm is proposed to take into account information about both hyperlinks and user behavior in the form of user traversals on hyperlinks.

We propose to apply the PageRate algorithm to a Web site link structure and link hierarchy respectively. Our proposition is based on two reasons. First, a link hierarchy includes structural links and excludes secondary links. A *structural link* shows conceptual relationship between the two pages A and B, is traversed frequently by users, and can be seen as a recommendation of page B by page A. A *secondary link* has *auxiliary* navigational functions, is rarely traversed by users, and may not be seen as a recommendation. Second, search on a link hierarchy can be integrated with the link hierarchy visualized in ONE and link prediction on the link hierarchy for adaptive Web site search and navigation.

Since extended anchor texts of a page are collective opinions about the content of the page by authors of the other pages, they can be less biased and more conclusive descriptions of the page than the page itself. We compare the extended anchor texts of a page to a user query to get a relevance-based ranking of the page. Relevance-based rankings of pages are combined with the PageRates of these pages as the overall rankings of these pages in search results.

5.2 Overview

As discussed in chapter 2, Page et al. [1998] proposed the PageRank algorithm to give authority-based rankings to pages. Pitkow et al. [2002] proposed to incorporate usage data in ranking Web pages. Ding and Chi [2000] classified user queries into exact queries and general queries in combining relevance-based, hyperlink-based, and usage based rankings in user search. Glover et al. [2002a and 2002b] claimed that the extended anchor texts of a page better summarize the content of the page. Built upon their work, we propose approaches for adapting search to user behavior on a Web site. First, we propose the PageRate algorithm, which takes into account information about both hyperlinks and user behavior in ranking Web pages. Second, we combine relevance-based rankings and PageRates of pages as overall rankings of these pages, which are used to rank search results.

5.3 PageRate Algorithm

User traversals on hyperlinks can be seen as their collective opinions in following these hyperlinks. Each hyperlink on a page does not necessarily have the same level of importance. It is obvious that a hyperlink in a prominent position, in large font, and in bold typeface is viewed by the Web page author as more important and will normally receive much more clicks from users than another hyperlink in an unnoticeable corner, in small font and in normal typeface.

We propose the *PageRate* algorithm, which takes into account information about user behavior in the form of user traversals on hyperlinks in ranking Web pages. As opposed to treating each hyperlink in a page equally in the PageRank algorithm, the more users have followed a hyperlink, the more important the page it links to. The

PageRate of a page A is biased toward the pages linked by frequently traversed hyperlinks in page A.

As an improvement of the PageRank algorithm, the PageRate algorithm also models a *random surfer's navigation* on the link structure of a Web site consisting of N nodes. A surfer is given a Web page at random. The surfer has two choices on each Web page. First, the surfer clicks each hyperlink on the page with a probability, which is the number of user traversals on the hyperlink divided by the sum of user traversals on all the out-links of the page multiplied by the damping factor d. Second, the surfer jumps to any page on the Web site link structure with an equal probability $(1\text{-}d)/N$. The surfer keeps clicking hyperlinks on the following pages. The PageRate of a page A is:

$$PR(A) = \frac{1-d}{N} + d \cdot \sum_{j=1}^{IN(A)} PR(P_j) \cdot \frac{n(P_j, A)}{\sum_{i=1}^{OUT(P_j)} n(P_j, P_i)} \qquad (5.1)$$

where $IN(A)$ is the number of in-links of page A, P_j is a page linking to page A, $n(P_j, A)$ is the number of user traversals on the hyperlink from page P_j to page A, $OUT(P_j)$ is the number of out-links of page P_j, and $\sum_{i=1}^{OUT(P_j)} n(P_j, P_i)$ is the sum of user traversals on all the out-links of page P_j.

Writing down Equation 5.1 for all N nodes, we get the vector v of the PageRates of pages:

$$v = (1-d) \cdot e + d \cdot (Q^T \times v) \qquad (5.2)$$

where e is an $1 \times N$ vector such that $e = (1/N, \ldots, 1/N)$, and Q is the transition matrix of a Markov chain model. In Q, the probability of following a hyperlink in a page is the number of user traversals on the hyperlink divided by the sum of user traversals on all the out-links of the page.

Equation 5.2 can be written in the form of a new transition matrix Q' that defines v:

$$v = (d \cdot Q^T + (1-d) \cdot M) \cdot v = (Q^{'})^T \cdot v \qquad (5.3)$$

where M is an $N \times N$ matrix such that $\forall i, j$, $M_{i,j} = 1/N$, and $Q^{'} = d \cdot Q + (1-d) \cdot M$, v is the *eigenvector* of the transition matrix $Q^{'}$, and also the *stationary distribution* of a Markov chain model, which has $Q^{'}$ as its transition matrix. The random surfing is equivalent to a random walk on the link structure. PageRate of a page is proportional to the frequency with which a random surfer will visit the page.

We apply the PageRate algorithm to a Web site link structure and link hierarchy, respectively. For example, Figure 5.1 shows the Web site link structure used for link prediction in chapter 4.

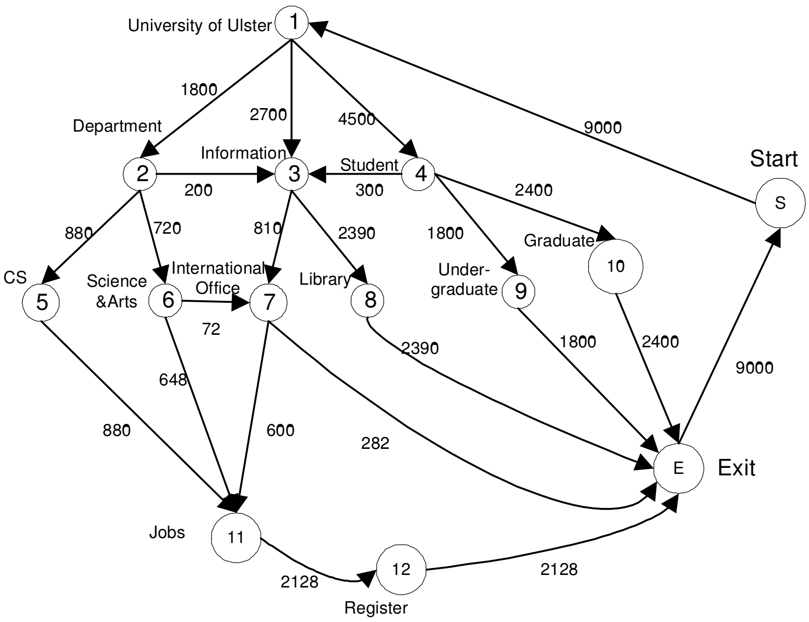


Figure 5.1: A Web site link structure with "Exit" and "Start" nodes.

We set $d = 0.5$. According to Equation 5.1, PageRate of page 7 is:

$$PR(7) = \frac{1-0.5}{14} + 0.5 \cdot \left\{ PR(3) \cdot \frac{810}{810+2390} + PR(6) \cdot \frac{72}{72+648} \right\}$$

The transition matrix Q is shown in Figure 5.2.

Page \ *Page*	1	2	3	4	5	6	7	8	9	10	11	12	*Exit*	*Start*
1		0.2	0.3	0.5										
2			0.111		0.489	0.4								
3							0.253	0.747						
4			0.067						0.4	0.533				
5											1.0			
6							0.1				0.9			
7											0.68		0.32	
8													1.0	
9													1.0	
10													1.0	
11												1.0		
12													1.0	
Exit														1.0
Start	1.0													

Figure 5.2: A transition matrix Q of a Markov chain on the Web site link structure in Figure 5.1.

According to Equation 5.3, $Q^{'} = 0.5 \cdot Q + (1 - 0.5) \cdot M$. The transition matrix $Q^{'}$ is shown in Figure 5.3.

Page \ *Page*	1	2	3	4	5	6	7	8	9	10	11	12	*Exit*	*Start*
1	0.036	0.136	0.186	0.286	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
2	0.036	0.036	0.092	0.036	0.281	0.236	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
3	0.036	0.036	0.036	0.036	0.036	0.036	0.163	0.41	0.036	0.036	0.036	0.036	0.036	0.036
4	0.036	0.036	0.07	0.036	0.036	0.036	0.036	0.036	0.236	0.303	0.036	0.036	0.036	0.036
5	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036	0.036	0.036
6	0.036	0.036	0.036	0.036	0.036	0.036	0.086	0.036	0.036	0.036	0.486	0.036	0.036	0.036
7	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.376	0.036	0.196	0.036
8	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036
9	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036
10	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036
11	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036	0.036
12	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036
Exit	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536
Start	0.536	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036

Figure 5.3: A transition matrix $Q^{'}$ of a Markov chain on the Web site link structure in Figure 5.1.

Figure 5.4 shows a link hierarchy of the Web site link structure in Figure 5.1.

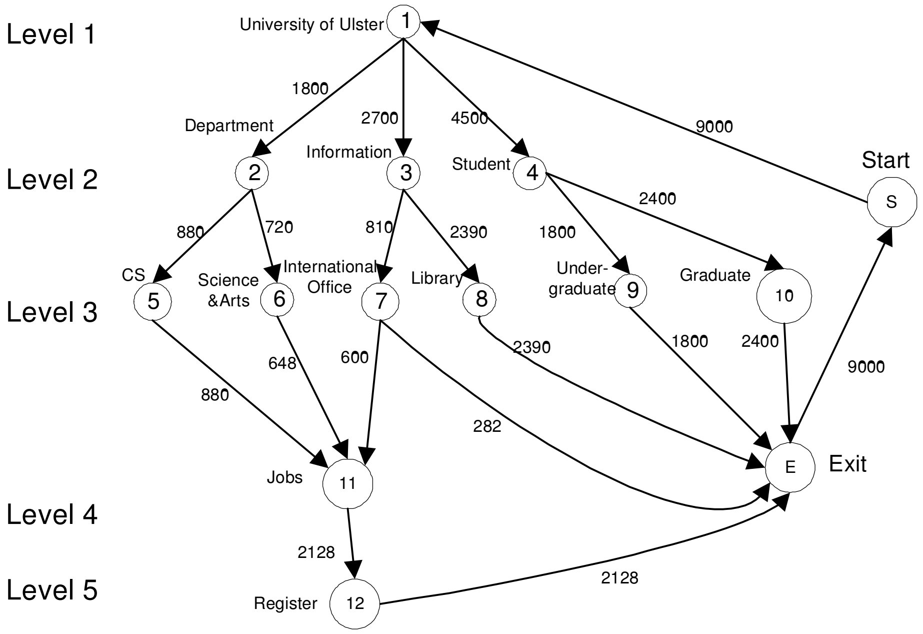


Figure 5.4: A link hierarchy of the Web site link structure in Figure 5.1 with "Exit" and "Start" nodes.

We set d =0.5. According to Equation 5.1, PageRate of page 7 is:

$$PR(7) = \frac{1-0.5}{14} + 0.5 \cdot PR(3) \cdot \frac{810}{810+2390}$$

The transition matrix Q is shown in Figure 5.5.

*Page**Page*	1,1	2,2	3,2	4,2	5,3	6,3	7,3	8,3	9,3	10,3	11,4	12,5	*Exit*	*Start*
1,1		0.2	0.3	0.5										
2,2					0.55	0.45								
3,2							0.253	0.747						
4,2									0.429	0.571				
5,3											1.0			
6,3											1.0			
7,3											0.68		0.32	
8,3													1.0	
9,3													1.0	
10,3													1.0	
11,4												1.0		
12,5													1.0	
Exit														1.0
Start	1.0													

Figure 5.5: The transition matrix Q of a Markov chain on the link hierarchy in Figure 5.4.

According to Equation 5.3, $Q^{'} = 0.5 \cdot Q + (1 - 0.5) \cdot M$. The transition matrix $Q^{'}$ is shown in Figure 5.6.

Page \ *Page*	1,1	2,2	3,2	4,2	5,3	6,3	7,3	8,3	9,3	10,3	11,4	12,5	*Exit*	*Start*
1,1	0.036	0.136	0.186	0.286	0.036	0.036	0.036	0036	0.036	0.036	0.036	0.036	0.036	0.036
2,2	0.036	0.036	0.036	0036	0.311	0.261	0.036	0036	0.036	0.036	0.036	0.036	0.036	0036
3,2	0.036	0.036	0.036	0.036	0.036	0.036	0.163	041	0.036	0.036	0.036	0.036	0.036	0.036
4,2	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0036	0.251	0.322	0.036	0.036	0.036	0.036
5,3	0.036	0.036	0.036	0.036	0036	0.036	0.036	0.036	0.036	0.036	0.536	0.036	0.036	0.036
6,3	0.036	0.036	0.036	0036	0.036	0.036	0.036	0036	0.036	0.036	0.536	0.036	0.036	0.036
7,3	0.036	0.036	0.036	0036	0.036	0.036	0.036	0036	0.036	0.036	0.376	0.036	0.196	0.036
8,3	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036
9,3	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0036	0.036	0.036	0.036	0.036	0.536	0.036
10,3	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0036	0.036	0.036	0036	0.036	0.536	0.036
11,4	0.036	0.036	0.036	0.036	0036	0.036	0.036	0.036	0.036	0.036	0.036	0.536	0.036	0.036
12,5	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0036	0.036	0.036	0036	0.036	0.536	0.036
Exit	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0036	0.536
Start	0.536	0.036	0.036	0.036	0036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036

Figure 5.6: The transition matrix $Q^{'}$ of a Markov chain on the link hierarchy in Figure 5.4.

We can use an iterative procedure to calculate the PageRates of pages in a Web site link structure and a link hierarchy, respectively. As the *initial state* we may simply set all the PageRates equal to $1/N$. Using Equation 5.3, the iteration process of the PageRate algorithm on a Web site link structure and a link hierarchy are shown in Table 5.1 and Table 5.2, respectively.

Table 5.1: Iterations of the PageRate algorithm on the Web site link structure in Figure 5.1, which reach a stationary distribution in nine rounds.

Iteration	*PR*(1)	*PR*(2)	*PR*(3)	*PR*(4)	*PR*(5)	*PR*(6)	*PR*(7)	*PR*(8)	*PR*(9)	*PR*(10)	*PR*(11)	*PR*(12)	*PR*(E)	*PR*(S)
0	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071
1	0.072	0.043	0.053	0.054	0.053	0.050	0.049	0.063	0.050	0055	0.128	0.072	0.190	0.072
2	0.072	0.043	0.051	0.054	0.047	0.045	0.045	0.056	0.047	0.050	0.102	0.100	0.163	0.131
3	0.102	0.432	0.051	0.054	0.047	0.045	0.045	0.055	0.047	0.050	0.095	0.083	0.163	0.121
4	0.095	0.046	0.055	0.061	0.047	0.045	0.045	0.055	0.047	0.050	0.095	0.083	0.163	0.121
5	0.096	0.045	0.055	0.060	0.047	0.045	0.045	0.057	0.048	0.052	0.095	0.083	0.161	0.117
6	0.095	0.046	0.055	0.060	0.047	0.045	0.045	0.0056	0.048	0.052	0.095	0.083	0.164	0.116
7	0.094	0.045	0.055	0.060	0.047	0.045	0.045	0.057	0.048	0.052	0.095	0.084	0.163	0.118
8	0.095	0.045	0.055	0.060	0.047	0.045	0.045	0.056	0.048	0.052	0.095	0.084	0.163	0.118
9	0.095	0.045	0.055	0.060	0.047	0.045	0.045	0.056	0.048	0.052	0.095	0.084	0.163	0.118

Table 5.2: Iterations of the PageRate algorithm on the link hierarchy in Figure 5.4, which reach a stationary distribution in ten rounds.

Iteration	*PR*(1)	*PR*(2)	*PR*(3)	*PR*(4)	*PR*(5)	*PR*(6)	*PR*(7)	*PR*(8)	*PR*(9)	*PR*(10)	*PR*(11)	*PR*(12)	*PR*(E)	*PR* (S)
0	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071	0.071
1	0.072	0.043	0.047	0.054	0.053	0.050	0.045	0.063	0.050	0055	0.128	0.072	0.190	0.072
2	0.072	0.043	0.047	0.054	0.047	0.045	0.042	0.053	0.047	0.050	0.101	0.100	0.163	0.131
3	0.102	0.043	0.047	0.054	0.047	0.045	0.042	0.053	0.047	0.050	0.094	0.086	0.168	0.117
4	0.095	0.046	0.051	0.061	0.047	0.045	0.042	0.053	0.047	0.050	0.093	0.083	0.161	0.120
5	0.096	0.045	0.050	0.060	0.047	0.045	0.042	0.055	0.048	0.052	0.094	0.083	0.159	0.117
6	0.094	0.046	0.050	0.060	0.047	0.045	0.042	0.055	0.048	0.052	0.094	0.083	0.162	0.115
7	0.094	0.045	0.050	0.060	0.047	0.045	0.042	0.054	0.048	0.052	0.094	0.083	0.161	0.117
8	0.095	0.045	0.050	0.059	0.047	0.045	0.042	0.055	0.048	0.052	0.094	0.083	0.162	0.117
9	0.094	0.045	0.050	0.060	0.047	0.045	0.042	0.055	0.048	0.052	0.094	0.083	0.161	0.117
10	0.094	0.045	0.050	0.060	0.047	0.045	0.042	0.055	0.048	0.052	0.094	0.083	0.161	0.117

5.4 Combining PageRates with Relevance-Based Rankings

In this section, we use feature vectors to represent user queries and pages, respectively. We compare two feature vectors representing a Web page and a query, respectively, to get a relevance-based ranking of the page. The PageRates of pages are combined with their relevance-based rankings as the overall rankings of these pages to rank search results.

We use the *TF-IDF weight* [Hand et al. 2001] in the feature vectors of pages. *TF* stands for feature frequency and means that the weight of each feature vector is the frequency with which the feature occurs. This has the effect of *increasing* the weight on features that occur frequently in a given page. However, if a feature occurs frequently in many pages, then using TF weights may have *little discriminative power* [Hand et al. 2001]. The *inverse-document-frequency (IDF) weight* helps to improve discrimination. For a collection of N pages, the IDF weight of feature j is defined as $\log(N / n_j)$, i.e., the log of the inverse of the fraction of pages in the collection that contain feature j. The IDF weight favors features that occur in relatively few pages. The TF-IDF weight is the product of TF and IDF for a feature in a page.

Since extended anchor texts of a page are collective opinions about the contents of the page, they can be less biased and more conclusive descriptions of the page than the page itself. Glover et al. [2002a and 2002b] claimed that the extended anchor texts of a page better summarize the contents of the page since people providing them are interested in the page. Hodgson [2001] showed that anchor texts of a Web page are accurate conceptual descriptions of the contents of the page.

We aggregate the extended anchor texts of a page for a feature vector that represents the page. All the words and phrases in the extended anchor texts of pages are considered as *candidate features*. We perform *thresholding*, by removing those rare words and phrases that do not occur frequently in these extended anchor texts. We then get a feature set consisting of M features, f_j, $1 \leq j \leq M$. Each extended anchor text E is represented as a feature vector, $E = (w_1, w_2, ..., w_M)$, where w_j is a *TF-IDF weight* if E contains feature f_j, and $w_j = 0$ otherwise.

Some Web pages may have multiple extended anchor texts. Glover et al[2002a and 2002b] treated the feature vector from each extended anchor text of a page equally in aggregating a feature vector for the page. We instead weight the feature vector from each extended anchor text of a page by the*in-link strength* as defined in chapter 3 in aggregating a feature vector for the page,$P = (p_1, ..., p_T)$.

A user query is also represented as a feature vector. Features, which do not occur in the query, are implicitly assigned zero weights. Individual weights can be assigned by user to indicate the relative importance of each feature. Let$Q = (q_1, ..., q_T)$ be a feature vector of the query. The *relevance-based ranking* $RR(P,Q)$, of page P in terms of distance to query Q is defined as the cosine distance between the two feature vectors as follows.

$$RR(P,Q) = \frac{\sum_{i=1}^{T} p_i \cdot q_i}{\sqrt{\sum_{i=1}^{T} p_i^2 \cdot \sum_{i=1}^{T} q_i^2}} \qquad (5.4)$$

This is the *cosine* of the *angle* between the two feature vectors and reflects similarity in terms of the *relative distribution* of their feature components. In order to ensure a certain level of relevancy of search results to a user query, we set a threshold on the relevance-based rankings of pages, i.e., only pages having relevance-based rankings above the threshold can be considered as search results. Relevance-based rankings of pages are integrated with PageRates of these pages as the overall rankings of them to rank search results in terms of both their *relevancy* and *authority*. Given a user query Q and a Web page P.

$$R(P,Q) = a \cdot RR(P,Q) + b \cdot PR(P,Q) \tag{5.5}$$

where $R(P,Q)$ is the overall ranking of page P, RR is the relevance-based ranking of page P, PR is the PageRate of page P, and a, b are weights indicating the importance of relevance-based ranking and authority-based ranking, respectively.

We classify user queries into *exact queries* and *general queries*. In an exact query, users know exactly what contents they are searching for. So the relevance-based ranking of a page should take a large portion in the overall ranking of the page, while the PageRate of the page takes a relatively small portion. In a general query, users usually do not have too much knowledge on the query topic, so they only use very *few* keywords and are more likely to see the pages other people visited and judged to be relevant. So the PageRate of a page should be more important than in the former case. For these two kinds of queries, we assign different values of a, b in Equation 5.5. Users can choose to specify the kind of query explicitly. We use some *heuristics* to infer the type of user query. If the number of query keywords is above a threshold, it is considered to be an *exact query*. Otherwise it is considered to be a *general query*. Intuitively it is true because users could describe more when they have more knowledge about what information they are searching for.

For example, a user issues a query "jobs" to search the Web site link structure in Figure 5.1. It is considered to be a *general query*, we set $a = b = 0.5$ in Equation 5.5. Suppose we have a feature set consisting of 15 words and phrases, and the tenth feature represents "jobs". We use TF-IDF weight in feature vectors. We suppose the fifth and tenth features occur in the extended anchor text of page 11 in page 5. The fifth feature occurs in the extended anchor texts of five pages, and the tenth feature occurs in the extended anchor texts of only one page, i.e., page 11. We get the feature vector representing the extended anchor text of page 11 in page 5 as:

$$E_{11,5} = \begin{vmatrix} 0 & 0 & 0 & 0 & \log(14/5) & 0 & 0 & 0 & 0 & \log 14 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

$$=\left|0 \quad 0 \quad 0 \quad 0 \quad 0.447 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.146 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

Similarly, we get the feature vectors representing the extended anchor texts of page 11 in page 6 and 7, respectively.

$$E_{11,6}=\left|0 \quad 0 \quad \log(14/6) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \log 14 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

$$=\left|0 \quad 0 \quad 0.368 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.146 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

$$E_{11,7}=\left|0 \quad 0 \quad 0 \quad \log(14/7) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \log 14 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

$$=\left|0 \quad 0 \quad 0 \quad 0.301 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.146 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

We use *in-link strengths* on the three links to page 11 to weight the feature vectors of the extended anchor texts in aggregating a feature vector to represent page 11.

$$P_{11}=\frac{880}{880+648+600}\cdot E_{11,5}+\frac{648}{880+648+600}\cdot E_{11,6}+\frac{600}{880+648+600}\cdot E_{11,7}$$

$$=\left|0 \quad 0 \quad 0.112 \quad 0.085 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1.146 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

The feature vector representing the user query is:

$$Q=\left|0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right|$$

According to Equation 5.4, the cosine distance between P_{11} and Q is:

$$CR(P_{11},Q)=\frac{1.146}{\sqrt{1.333\cdot 1}}=0.993$$

The PageRate of page 11 in Table 5.1 is 0.095. According to Equation 5.5, the overall ranking of page 11 to the user query is:

$$R(P_{11},Q)=0.5\cdot 0.993+0.5\cdot 0.095=0.544$$

5.5 Comparisons with Related Work

First, we compare the PageRate algorithm with the PageRank algorithm in giving *authority-based rankings* to pages. Both the PageRate and PageRank algorithms model a *random surfer's navigation* on the Web link structure. A surfer is given a page at random. With probability d, the *damping factor*, the surfer decides to choose a hyperlink in each page with a certain probability. With probability $(1d)$, the surfer jumps to a random page on the Web link structure. The surfer keeps clicking hyperlinks on following pages.

The two algorithms differ in terms of the probability with which the random surfer chooses a hyperlink in each page. In the PageRank algorithm, the surfer chooses a hyperlink in each page with an equal probability decided by the number of hyperlinks on the page. In the PageRate algorithm, the surfer chooses a hyperlink in each page with a probability, which is the number of user traversals on the hyperlink divided by the sum of user traversals on all the hyperlinks in the page multiplied by the damping factor d.

The random surfer's navigation is equivalent to a *random walk* on the Web link structure and can be modeled by a Markov chain. In determining the one-step transition probabilities from one page to other pages in the transition matrix of a Markov chain model, the PageRank algorithm only takes into account the number of hyperlinks in the page, while the PageRate algorithm also takes into account user traversals on hyperlinks in the page. Both PageRates and PageRanks of pages can be understood as the *steady-state probability distribution* of the two Markov chain models, respectively.

Second, we compare our method with Glover et al.'s method in aggregating feature vectors for pages. Glover et al. [2002a and 2002b] treated the feature vector representing each extended anchor text of a page equally when synthesizing a feature vector to represent the page. Given a page, we instead weight the feature vector from each extended anchor text of a page by the in-link strength as defined in chapter 3 in aggregating a feature vector for the page.

5.6 Summary

In this chapter, we presented our approaches to user search ranking. We proposed the PageRate algorithm, which takes into account information about both hyperlinks and user behavior in giving an authority-based ranking to each page in a Web site link structure and link hierarchy, respectively. We proposed to use feature vectors extracted from extended anchor texts of a page to represent a page. Two feature vectors representing a user query and a page, respectively, are compared to get a relevance-based ranking of the page, which is combined with the PageRate of the page as the overall ranking of the page in ranking search results. Search results are used to help users find desired information on the Web site effectively and efficiently.

User behavior in the form of traversals on hyperlinks is used to estimate one-step transition probabilities between pages, which are used in the PageRate algorithm. If user behavior changes, our approaches are influenced in two aspects. First, transition probabilities between pages change. Second, a different link hierarchy is constructed. If we use Web log files recording user behavior changes over time to estimate transition probabilities, different PageRates are given to pages. Different feature vectors may be extracted from extended anchor texts to represent pages. Relevance-based rankings of pages change as well. Thus the overall rankings of pages may change the ordering of pages in search results. In our future work, we intend to study the influence of user behavior changes on the PageRate algorithm and user search. We also plan to apply our approach to other Web sites.

Chapter 6

EXPERIMENTAL EVALUATION

In this chapter, we present our experiments on the University of Ulster Web site to evaluate our approaches to adaptive Web site navigation and search presented in chapter 3, 4, and 5 respectively. The chapter is organized as follows. In section 6.1, we give an overview of a prototype called Online Navigation Explorer (ONE), which has been used as the user interface for the evaluation. In section 6.2, we evaluate our approaches to mining link hierarchies and conceptual link hierarchies described in chapter 3. In section 6.3, we evaluate our approaches to link prediction using Markov chain models presented in chapter 4. In section 6.4, we evaluate our approaches to ranking search results in response to users' keyword-based queries proposed in chapter 5. Finally we conclude in section 6.5.

6.1 Overview of Online Navigation Explorer (ONE)

To study the effectiveness and efficiency of our approaches to adaptive Web site navigation and search, we conducted an evaluation using a prototype implemented as a Visual Basic 6.0 program called*Online Navigation Explorer*(*ONE*). *ONE* consists of five components, namely, components for visualizing link hierarchies or conceptual link hierarchies, browsing pages, link prediction using Markov chain models, search by users' keyword-based queries, and options and controls for user evaluation.

ONE supports seven kinds of interfaces, namely, interfaces that support browsing alone and browsing assisted by one of the following facilities:

- **the visualized hierarchy**[9]. Using the hierarchy, users can control their navigation and thus are not confined to following only hyperlinks in each page. They can move up and down in the hierarchy or jump from one page or cluster to another page or cluster. The hierarchy can help users understand the relationships between the pages they have visited and their current locations in the context of different conceptual levels consisting of pages and clusters in the hierarchy. Based on these understandings, they can decide where they can go next.

- **link prediction.** Most probably to-be-visited (MPT) pages within the next*n* steps are predicted. In the next step, users are not limited to choosing a page linked by the current page, and can directly jump to a page, which contains desired information but is many links away from the current page.

- **link prediction and the visualized hierarchy**MPT pages and cluster within the next *n* steps and guided paths are highlighted on the hierarchy and also presented to users. Guided paths help users find a path through the hierarchy of a Web site in order to find desired information. By visualizing link prediction results on the hierarchy, we can help users explore deeper into the hierarchy to find desired information.

- **search**. Search results are presented to users in response to their keyword-based queries. Search is suitable for information seeking tasks that can be easily described using keywords. User queries are classified into general queries and specific queries in combining relevance-based rankings with quality-based rankings of pages.

- **search and the visualized hierarchy.** Search results are highlighted on the hierarchy and also presented to users. By highlighting search results on the

[9] A hierarchy is either a link hierarchy or conceptual link hierarchy.

hierarchy, we can help users understand the search results in the context of different conceptual levels consisting of pages and clusters in the hierarchy.

- **search, the visualized hierarchy, and link prediction.** Link prediction results and search results are highlighted on the hierarchy and also presented to users. Users are allowed to switch smoothly between search and navigation, and to locate desired information matching their complex information searching tasks.

6.2 Evaluation of Approaches to Mining Link Hierarchies and Conceptual Link Hierarchies

In this section, we evaluate our approaches to mining *link hierarchies* and *conceptual link hierarchies*. The goal of our experiments is three-fold. First, we want to prove that our method can put pages onto the conceptual levels of a link hierarchy more accurately than both the breadth-first search method and the shortest weighted path method in link hierarchy construction. Second, we want to show that *PageCluster* can cluster conceptually related pages more accurately than the bibliographic analysis method. The constructed conceptual link hierarchy, where pages on each conceptual level of the link hierarchy have been clustered, is more compact than the link hierarchy. Third, the conceptual link hierarchy visualized in ONE can help users find information more effectively and efficiently than browsing alone as the task of finding information becomes less specific and involves more Web pages on multiple conceptual levels.

6.2.1 Evaluation of the Link Hierarchy Construction Method

The link structure of the University of Ulster Web site constructed from an ECLF format log file consists of 3,546 pages and 3,953 links. The log file is 371 MB in size and contains 2,193,998 requests. To ensure that collective user behavior in using the links are reflected in the link structure, for a link to be included in the link hierarchy it has to be traversed by at least three different users[10].

We used the breadth-first search method, the shortest weighted path method, and our own method to construct three different link hierarchies.For the shortest weighted path method, we assign weights on links in inverse proportion to the numbers of user traversals on the links. We included structural links and discarded secondary links in the link hierarchies constructed using the breadth-first search method and the shortest weighted path method respectively.We then asked the Web master, and an experienced user who has used the university Web site extensively for more than two years, to evaluate each of the three link hierarchies on the basis of theirunderstanding of the hierarchical organization of the Web site. They conducted their evaluation separately and were not told which method was used to construct each link hierarchy to avoid any bias towards a particular method. For each page on a conceptual level of the link hierarchy, they first indicated another page that is the main parent of the page in the hierarchical organization of the Web site. They then checked whether the indicated page is actually both on the adjacent higher conceptual level of the link hierarchy and linked to the page. An error occurred if this is not the case. The evaluation results are shown in Table 6.1.

[10] We assume each originating machine corresponds to a different user. These may not always be true when proxy servers exist. But in the absence of user tracking software, the method can still provide quite reliable results.

Table 6.1: Results of evaluating three link hierarchies constructed using our method, the breadth-first search method, and the shortest weighted path method.

	Level	2	3	4	5	6	7	8	9	10
Our method	Number of pages	23	172	852	540	170	151	64	46	36
	Number of structural links from adjacent higher level	23	233	870	548	175	156	66	46	36
	Number of errors identified by Web master	0	1	9	5	4	5	3	3	2
	Number of errors identified by user	0	2	8	6	4	4	2	2	2
Breadth-first search method	Number of pages	23	174	885	535	168	148	62	44	32
	Number of structural links from adjacent higher level	23	241	893	540	175	152	64	44	32
	Number of errors identified by Web master	0	3	32	9	13	11	20	8	9
	Number of errors identified by user	0	4	30	8	14	12	21	7	9
Shortest weighted path method	Number of pages	23	173	879	533	169	150	62	45	35
	Number of structural links from adjacent higher level	23	235	887	537	172	154	64	45	35
	Number of errors identified by Web master	0	2	27	8	12	9	18	5	5
	Number of errors identified by user	0	3	26	9	12	11	19	5	4

As shown in Table 6.1, our method outperformsboth the breadth-first search method and the shortest weighted path method in terms of the numbers of errors identified by both the Web master and the experienced user.

6.2.2 Evaluation of the PageCluster Algorithm

We clustered pages on each conceptual level of the link hierarchy constructed using our method. We set the weights for uncommon out-links and in-links and common out-links and in-links as w_U=1.5 and w_C=0.5 respectively. We set both the in-link and out-link similarity thresholds as 0.2 for any conceptual level of less than 200 pages, and 0.3 for any conceptual level of more than 200 pages. We also used the bibliographic analysis method to cluster pages on eachconceptual level for comparison. We used co-citation and coupling as similarity measures instead of in-link and out-link similarities for navigation clustering, and co-citation instead of in-link similarity for category clustering. We presented the clusters to the Web master and the experienced user for evaluation. They pointed out which pages have been misclustered based on their understanding of the organization of the Web site. They conducted their evaluation separately and were not told which method was used to cluster the pages on each conceptual level to avoid any bias towards a particular method. The evaluation of the clustering results of the PageCluster algorithm and the bibliographic analysis methods shown in Table 6.2. We can see that the PageCluster algorithm outperformed the bibliographic analysis method in terms of the number of misclustered pages pointed out by both the Web master and the experienced user.

Table 6.2: The evaluation of the clustering results of the PageCluster algorithm and the bibliographic analysis method on each conceptual level of the link hierarchy.

	Level	2	3	4	5	6	7	8	9	10
PageCluster	Number of NCs (Navigation Clusters)	2	7	25	15	3	2	1	0	0
	Average size	7	4	3.8	4.8	4	3	3	0	0
	Number of misclassified pages identified by Web master	0	5	11	11	1	0	0	0	0
	Number of misclassified pages identified by user	0	5	11	10	0	1	0	0	0
	Number of CCs (Category Clusters)	1	11	52	11	6	4	2	1	0
	Average size	9	8	6.5	9	6.5	5	4	2	0
	Number of misclassified pages identified by Web master	0	6	38	13	2	2	0	0	0
	Number of misclassified pages identified by user	0	5	35	12	2	2	0	0	0
	Number of unclustered pages	0	56	419	369	119	125	53	44	36
Bibliographic analysis method	Number of NCs	2	8	27	16	4	3	1	0	0
	Average size	7	4	4	5.625	4	3	3	0	0
	Number of misclassified pages identified by Web master	3	14	38	27	5	3	0	0	0
	Number of misclassified pages identified by user	4	15	37	26	6	4	1	0	0
	Number of CCs	1	12	58	14	8	5	2	1	0
	Average size	9	7.5	6.5	9	6.5	5	4	2	0
	Number of misclassified pages identified by Web master	0	25	61	32	17	7	2	0	0
	Number of misclassified pages identified by user	0	26	63	33	16	8	3	0	0
	Number of unclustered pages	0	52	399	321	98	113	52	41	31

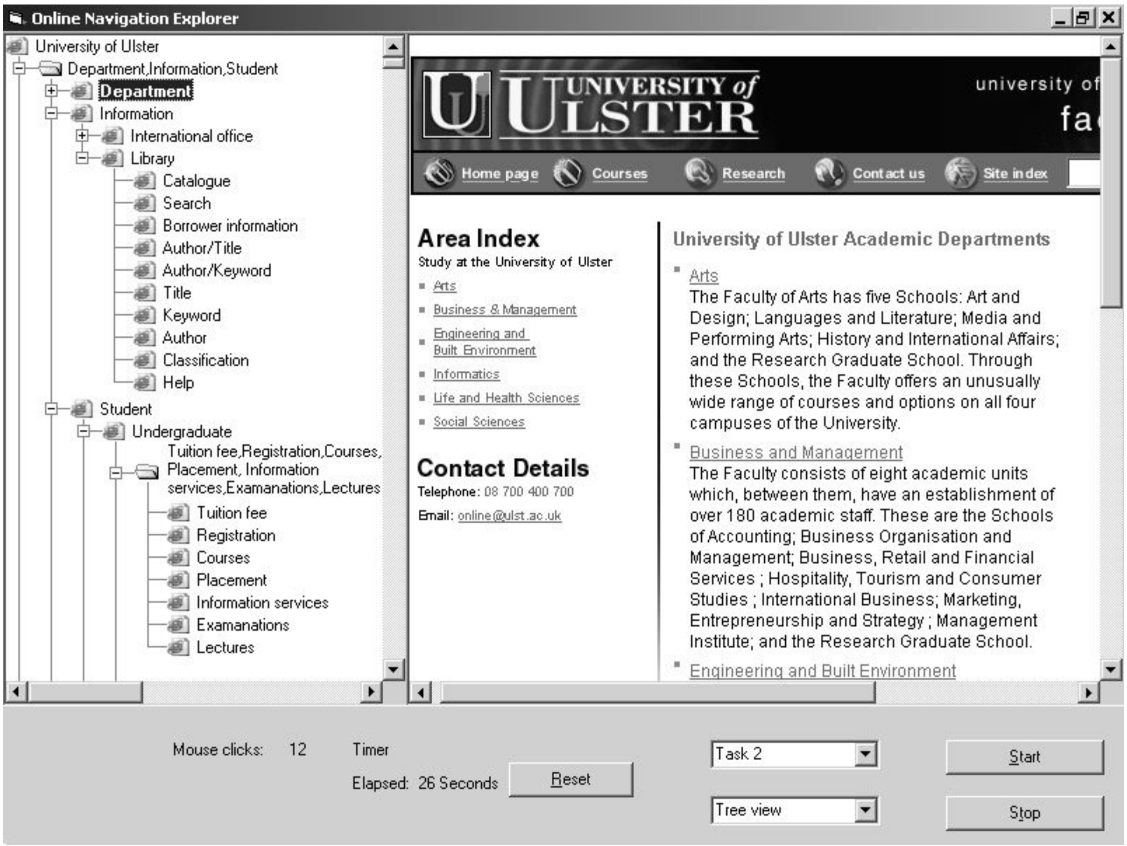
We used extended anchor texts to synthesize titles for pages and clusters. 102 features have been selected to form the feature vectors. The 5 features with the top five highest aggregated link strengths in the feature vector of a page or cluster are used as the title of the page or cluster. We next used the clusters and unclustered pages to construct a conceptual link hierarchy As shown in Table 6.3, the numbers of nodes and links in the conceptual link hierarchy have both been dramatically reduced compared with the link hierarchy. The conceptual link hierarchy provides a more compact view of the Web site.

Table 6.3: The results of conceptual link hierarchy construction.

Level	2	3	4	5	6	7	8	9	10
Number of nodes	3	74	496	395	128	131	56	45	36
Node reduction rate (%)	87	57	42	27	25	13	13	2	0
Number of in-links	3	81	510	403	132	134	58	45	36
Link reduction rate (%)	87	65	41	26	25	14	12	2	0

6.2.3 *Evaluation of Visualizing Conceptual Link Hierarchies*

As shown in Figure 6.1, we visualize a conceptual link hierarchy or link hierarchy as a tree, which is similar to the way the file system is visualized in the Microsoft Windows.



- University of Ulster
  - Department,Information,Student
    - Department
    - Information
      - International office
      - Library
        - Catalogue
        - Search
        - Borrower information
        - Author/Title
        - Author/Keyword
        - Title
        - Keyword
        - Author
        - Classification
        - Help
    - Student
      - Undergraduate
        - Tuition fee,Registration,Courses, Placement, Information services,Examanations,Lectures
          - Tuition fee
          - Registration
          - Courses
          - Placement
          - Information services
          - Examanations
          - Lectures

Area Index

Study at the University of Ulster

- Arts
- Business & Management
- Engineering and Built Environment
- Informatics
- Life and Health Sciences
- Social Sciences

Contact Details

Telephone: 08 700 400 700

Email: online@ulst.ac.uk

University of Ulster Academic Departments

- Arts
  The Faculty of Arts has five Schools: Art and Design; Languages and Literature; Media and Performing Arts; History and International Affairs; and the Research Graduate School. Through these Schools, the Faculty offers an unusually wide range of courses and options on all four campuses of the University.
- Business and Management
  The Faculty consists of eight academic units which, between them, have an establishment of over 180 academic staff. These are the Schools of Accounting; Business Organisation and Management; Business, Retail and Financial Services ; Hospitality, Tourism and Consumer Studies ; International Business; Marketing, Entrepreneurship and Strategy ; Management Institute; and the Research Graduate School.
- Engineering and Built Environment

Mouse clicks: 12

Timer

Elapsed: 26 Seconds

Task 2

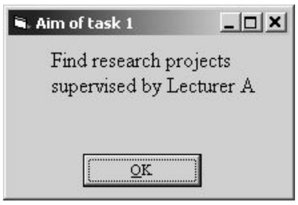
Tree view

Figure 6.1: A conceptual link hierarchy visualized in ONE

On the left side of the window is aleft-to-right tree representing the conceptual link hierarchy. The icon 📄 represents a page and 📂 represents a cluster. A node having children can be opened or closed by a mouse-click to view or hide its children. On the right side of the window, the current page is shown in the browser. Clicking on a page in the hierarchy changes the browsed page. Clicking on a link on the page to view another page in the browser changes the focused page in the hierarchy.

Nine tasks of three types can be selected. The three types of tasks are of increased complexity, and each type of tasks consists of three separate tasks of the same complexity. The three tasks of the first type are to find three particular lecturers' home pages. The three tasks of the second type are to find registration information, tuition fees, and accommodation information respectively for both graduate and undergraduate students for the coming new semester. The three tasks of the third type are to find as many as possible research projects supervised by three particular lecturers[11] respectively.

On the bottom of the window, a user can select a task to perform. When a task is selected, a window displaying the detail of the task is shown as in Figure 6.2.

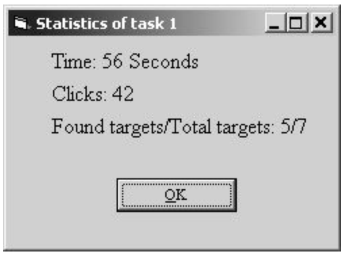


Find research projects supervised by Lecturer A

Figure 6.2: A window showing the detail of a task

As shown in Figure 6.1, a user can select the two methods, which are browsing assisted by a hierarchy and browsing alone. Users click the start button once they have selected a task and a method. Once the user has found answers to the task by opening pages containing the target information in the browser, they click the stop button and ONE measures the time taken and number of clicks used by the user to complete the task. A window that pops up showing the detail of the task completion is shown in Figure 6.3.

[11] These three lecturers are different from the three lecturers in the three tasks of the first type.



Statistics of task 1

Time: 56 Seconds

Clicks: 42

Found targets/Total targets: 5/7

Figure 6.3: A window showing the detail of *a completed task.*

Users can choose two tasks of each type for browsing alone and browsing assisted by a hierarchy, respectively. As the task becomes more complex and less specific, most of the time, the participants performed better with the hierarchy than without it in terms of the success rate of finding targets as well as the time and the number of clicks used.

Table 6.4: The results of evaluating the effectiveness and efficiency of browsing assisted by a hierarchy and browsing alone. The effectiveness is measured by the number of pages found versus the total number of target pages. The efficiency is measured by both the time and the number of clicks used by a subject performing a particular task.

		Tree view	Found/Target	Time (secs.)	Number of clicks
Task 1	Subject 1	Yes	1/1	30	2
		No	1/1	33	2
	Subject 2	Yes	1/1	42	2
		No	1/1	40	3
	Subject 3	Yes	1/1	25	2
		No	1/1	35	2
	Subject 4	Yes	1/1	20	2
		No	1/1	33	3
Task 2	Subject 1	Yes	2/2	42	4
		No	2/2	66	5
	Subject 2	Yes	2/2	67	5
		No	1/2	101	8
	Subject 3	Yes	2/2	35	4
		No	2/2	49	7
	Subject 4	Yes	2/2	25	4
		No	1/2	39	6
Task 3	Subject 1	Yes	12/12	286	17
		No	10/12	790	31
	Subject 2	Yes	10/12	389	19
		No	3/12	1036	48
	Subject 3	Yes	8/12	456	23
		No	1/12	789	32
	Subject 4	Yes	12/12	195	15
		No	12/12	489	28

As shown in Table 6.4, hierarchies can help users find desired information more effectively and efficiently than browsing alone as the task of finding information becomes less specific and involves more pages on different conceptual levels. User feedback also suggested that the conceptual link hierarchy had given them a clearer view of their current locations on the Web site, and better understanding of the conceptual relationships between clusters and pages. One user wrote, “This tool is very cool. I have used it to find more high-quality information about lecturers and their research projects quicker than using Web browsers. I can directly click links to pages I am interested in on the tree, not like having to find these links lurking somewhere on the page using Web browsers.”

6.3 Evaluation of Approaches to Link Prediction Using Markov Chain Models

In this section, we evaluate our approaches to link prediction using Markov chain models. The goal of our experiments is threefold. First, we want to show that the compression algorithm proposed by Spears [1998] can compress the transition matrix of a Markov chain model constructed from a Web site link structure (MMS) for efficient link prediction. Second, link prediction using MMSs can help users find desired information more effectively and efficiently than browsing alone. Third, link prediction using Markov chain models constructed from link hierarchies (MMHs) and conceptual link hierarchies (MMCs) can be integrated with visualized hierarchy in order to help users find desired information more effectively and efficiently than link prediction alone.

6.3.1 Evaluation of the Transition Matrix Compression Algorithm

We used the same log file used in Section 6.1 to construct a Web site link structure consisting of 2,173 pages and 3,187 links. The maximum number of traversals on a link in the link structure is 101,336, which is on the link from the “Start” node to the homepage of the Web site. The maximum and average numbers of links in a page in the link graph are 75 and 1.47 respectively. The maximum number of in-links of a page in the link graph is 57.

The transition matrix is 2175×2175 and very sparse. By setting six different thresholds for compression, we get the experimental results in Table 6.5.

Table 6.5: Compression results on a transition matrix.

ε	Compression Time (Minutes)	Size after compression	Percentage of states removed (%)
0.03	107	1627	25.2
0.05	110	1606	26.2
0.08	118	1579	27.4
0.12	122	1549	28.8
0.15	124	1542	29.1
0.17	126	1539	29.2

We can see that when ε increases, the matrix becomes harder to compress. For this matrix, we choose ε=0.15 for a good compression rate without significant errors. Experiments by Spears [1998] also show that a value of ε=0.15 yielded good compression with minimum errors. Now we calculateQ_c^2 and use the time spent as the benchmark for Q_c^m. Since we can repeatedly multiply Q_c^2 by Q_c to get Q_c^2, …, Q_c^{m-1}, Q_c^m, the time spent for computing Q_c^2,…, Q_c^{m-1}, Q_c^m can be estimated as the $(m-1)$ times of the time for Q_c^2. The experimental results of computation for Q_c^2 are summarized in Table 6.6.

Table 6.6: Experimental results forQ^2 and Q_c^2.

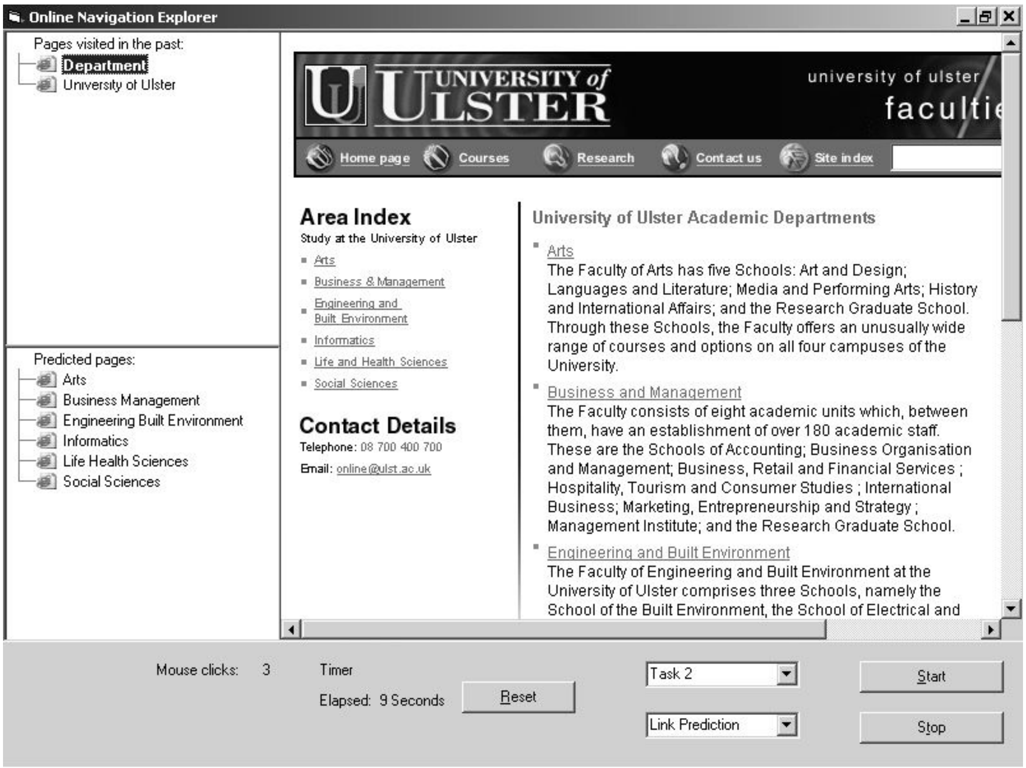
Matrix Dimension(^{2})	2175	1627	1606	1579	1549	1542	1539
Computation Time for Q^2 or Q_c^2 (Mins)	1483	618	592	561	529	521	518
Percentage of time saved (%)	N/A	58.3	60.1	62.1	64.3	64.9	65.1

We can see that the time needed for compression is compensated by the time saved in the computation for Q_c^2**.** When calculating Q^m, computational time can be further reduced. Q^2,…, Q^m can be computed off-line and stored for link prediction. So the response time is not an issue given the fast developing computational capability of the Web servers.

6.3.2 Evaluation of Link Prediction Using Markov Chain Models Constructed from Web Site Link Structures (MMSs)

We use the compressed transition matrix for link prediction. Link prediction results are presented in ONE to assist user navigation on the University of Ulster Web site. The average time needed for updating the prediction results is less than 30 seconds, so it is suitable for online navigation, given the response can be speeded up with the computational capability of current commercial Web sites. We selected m =5 and n =5 in link prediction to take into account five steps in a user's history in predicting five steps in the future. We computed Q^2 , …, Q^9 for link prediction.

As shown in Figure 6.4, ONE presents the last five visited pages as a user's history along with the predicted pages within the next five steps updated while browsing the Web site. Each time when a user requests a new page, probabilities of visiting other pages within the next five steps are re-calculated. Link prediction results are updated, i.e., pages with the highest probabilities in descending order are presented.



Pages visited in the past:
- Department
- University of Ulster

Predicted pages:
- Arts
- Business Management
- Engineering Built Environment
- Informatics
- Life Health Sciences
- Social Sciences

Area Index

Study at the University of Ulster
- Arts
- Business & Management
- Engineering and Built Environment
- Informatics
- Life and Health Sciences
- Social Sciences

Contact Details

Telephone: 08 700 400 700

Email: online@ulst.ac.uk

University of Ulster Academic Departments

- Arts
  The Faculty of Arts has five Schools: Art and Design; Languages and Literature; Media and Performing Arts; History and International Affairs; and the Research Graduate School. Through these Schools, the Faculty offers an unusually wide range of courses and options on all four campuses of the University.
- Business and Management
  The Faculty consists of eight academic units which, between them, have an establishment of over 180 academic staff. These are the Schools of Accounting; Business Organisation and Management; Business, Retail and Financial Services ; Hospitality, Tourism and Consumer Studies ; International Business; Marketing, Entrepreneurship and Strategy ; Management Institute; and the Research Graduate School.
- Engineering and Built Environment
  The Faculty of Engineering and Built Environment at the University of Ulster comprises three Schools, namely the School of the Built Environment, the School of Electrical and

Mouse clicks: 3

Timer

Elapsed: 9 Seconds

Task 2

Link Prediction

Figure 6.4: Link prediction in ONE.

The nine tasks in Section 6.2.3 are used for user study. Users can choose two tasks of each type for using the browser assisted by link prediction and the browser alone, respectively. As we can see in Table 6.7, most of the time, the participants performed better with link prediction than without it in terms of the success rate of finding targets as well as the time and the number of clicks used.

Table 6.7: The results of evaluating the effectiveness and efficiency of browsing assisted by link prediction and browsing alone.

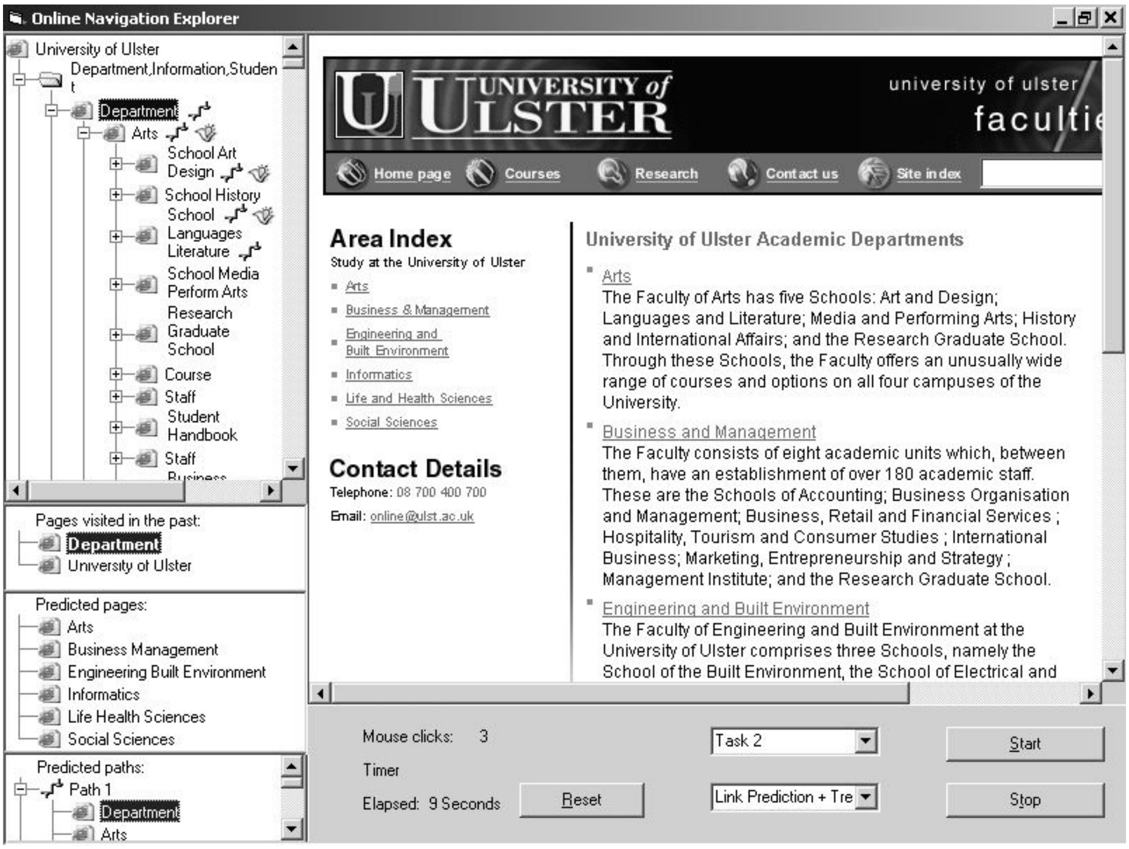
		Link prediction	Found/Target	Time (secs.)	Number of clicks
Task 1	Subject 1	Yes	1/1	30	2
		No	1/1	36	2
	Subject 2	Yes	1/1	26	2
		No	1/1	33	3
	Subject 3	Yes	1/1	23	2
		No	1/1	26	2
	Subject 4	Yes	1/1	29	3
		No	1/1	28	3
Task 2	Subject 1	Yes	2/2	56	5
		No	2/2	60	5
	Subject 2	Yes	2/2	58	6
		No	2/2	78	9
	Subject 3	Yes	2/2	60	5
		No	2/2	86	8
	Subject 4	Yes	2/2	53	5
		No	1/2	59	6
Task 3	Subject 1	Yes	9/12	723	29
		No	10/12	801	33
	Subject 2	Yes	9/12	1211	50
		No	7/12	1236	53
	Subject 3	Yes	11/12	522	25
		No	11/12	602	28
	Subject 4	Yes	8/12	673	35
		No	7/12	665	31

Link prediction can help users find desired information more effectively and efficiently than browsing alone. User feedback also suggested that link prediction results had helped them navigate to their desired information on the Web site more easily.

6.3.3 Evaluation of Link Prediction Using Markov Chain Models Constructed from Link Hierarchies (MMHs) and Conceptual Link Hierarchies (MMCs)

By visualizing link prediction results using MMHs and MMCs on a hierarchy, we can help users find information more effectively and efficiently than either visualized hierarchies alone or link prediction alone.

We selected $m=5$ and $n=5$ in link prediction to take into account five steps in a user's history in predicting five steps in the future. We computed Q^2, …, Q^9 for link prediction. Given a sequence of pages and clusters as a user's history, the most probably to-be-visited (MPT) pages and clusters within the next five steps are predicted, visualized on the hierarchy, and listed in descending order of their probabilities as shown in Figure 6.5. Given a user's current page or cluster, *guided paths* on the hierarchy are predicted, visualized on the hierarchy, and listed. The icon ☞ represents *predicted pages* and ⤴ represents pages on guided path. Each time when a user navigates to a new page, the user's history and link prediction results are updated and visualized.



Area Index
Study at the University of Ulster
- Arts
- Business & Management
- Engineering and Built Environment
- Informatics
- Life and Health Sciences
- Social Sciences

Contact Details
Telephone: 08 700 400 700
Email: online@ulst.ac.uk

University of Ulster Academic Departments

- Arts
The Faculty of Arts has five Schools: Art and Design; Languages and Literature; Media and Performing Arts; History and International Affairs; and the Research Graduate School. Through these Schools, the Faculty offers an unusually wide range of courses and options on all four campuses of the University.
- Business and Management
The Faculty consists of eight academic units which, between them, have an establishment of over 180 academic staff. These are the Schools of Accounting; Business Organisation and Management; Business, Retail and Financial Services ; Hospitality, Tourism and Consumer Studies ; International Business; Marketing, Entrepreneurship and Strategy ; Management Institute; and the Research Graduate School.
- Engineering and Built Environment
The Faculty of Engineering and Built Environment at the University of Ulster comprises three Schools, namely the School of the Built Environment, the School of Electrical and

Mouse clicks: 3
Timer
Elapsed: 9 Seconds
Task 2
Link Prediction + Tre

Figure 6.5: Integrating link prediction using a MMC with a conceptual link hierarchy in ONE.

The nine tasks in Section 6.2.3 are used for user study. The three tasks of each type are for link prediction using the MMH integrated with the visualized link hierarchy, the visualized link hierarchy alone, and link prediction using the MMH alone, respectively. As we can see in Table 6.8, most of the time, the participants performed better with link prediction integrated with the visualized link hierarchy than both the visualized link hierarchy alone and link prediction alone in terms of the success rate of finding targets as well as the time and the number of clicks used.

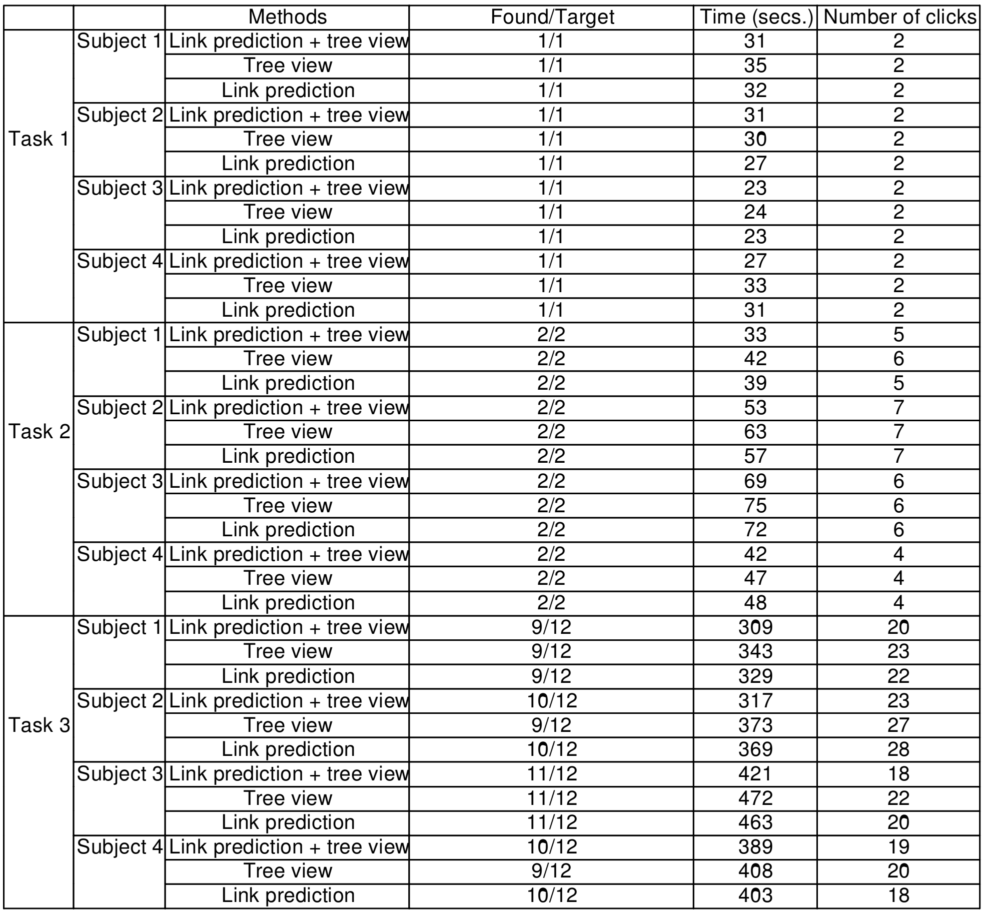
Table 6.8: The results of evaluating the effectiveness and efficiency of link prediction using a MMH integrated with a visualized link hierarchy, a visualized link hierarchy, and link prediction using a MMH, respectively.

		Methods	Found/Target	Time (secs.)	Number of clicks
Task 1	Subject 1	Link prediction + tree view	1/1	32	2
		Tree view	1/1	30	2
		Link prediction	1/1	31	2
	Subject 2	Link prediction + tree view	1/1	26	2
		Tree view	1/1	32	2
		Link prediction	1/1	33	2
	Subject 3	Link prediction + tree view	1/1	23	2
		Tree view	1/1	26	2
		Link prediction	1/1	27	2
	Subject 4	Link prediction + tree view	1/1	29	2
		Tree view	1/1	28	2
		Link prediction	1/1	27	2
Task 2	Subject 1	Link prediction + tree view	2/2	37	5
		Tree view	2/2	45	5
		Link prediction	2/2	52	5
	Subject 2	Link prediction + tree view	2/2	51	6
		Tree view	2/2	57	6
		Link prediction	2/2	69	8
	Subject 3	Link prediction + tree view	2/2	52	5
		Tree view	2/2	59	5
		Link prediction	2/2	69	8
	Subject 4	Link prediction + tree view	2/2	43	4
		Tree view	2/2	44	4
		Link prediction	1/2	47	5
Task 3	Subject 1	Link prediction + tree view	9/12	310	20
		Tree view	9/12	303	21
		Link prediction	8/12	412	27
	Subject 2	Link prediction + tree view	10/12	353	25
		Tree view	9/12	359	24
		Link prediction	7/12	443	30
	Subject 3	Link prediction + tree view	11/12	482	21
		Tree view	11/12	501	28
		Link prediction	8/12	623	37
	Subject 4	Link prediction + tree view	10/12	443	20
		Tree view	10/12	467	23
		Link prediction	7/12	532	36

User feedback also suggested that link prediction integrated with the visualized link hierarchy is more useful than both the visualized link hierarchy alone and link prediction alone in helping them find the desired information.

The three tasks of each type are then used for link prediction using the MMC integrated with the visualized conceptual link hierarchy, the visualized conceptual link hierarchy alone, and link prediction using the MMC alone, respectively. As we can see in Table 6.9, most of the time, the participants performed better with link prediction integrated with the visualized conceptual link hierarchy than both the conceptual link hierarchy alone and link prediction alone in terms of the success rate of finding targets as well as the time and the number of clicks used.

Table 6.9: The results of evaluating the effectiveness and efficiency of link prediction using a MMC integrated with a visualized conceptual link hierarchy, a visualized conceptual link hierarchy, and link prediction using a MMC, respectively.



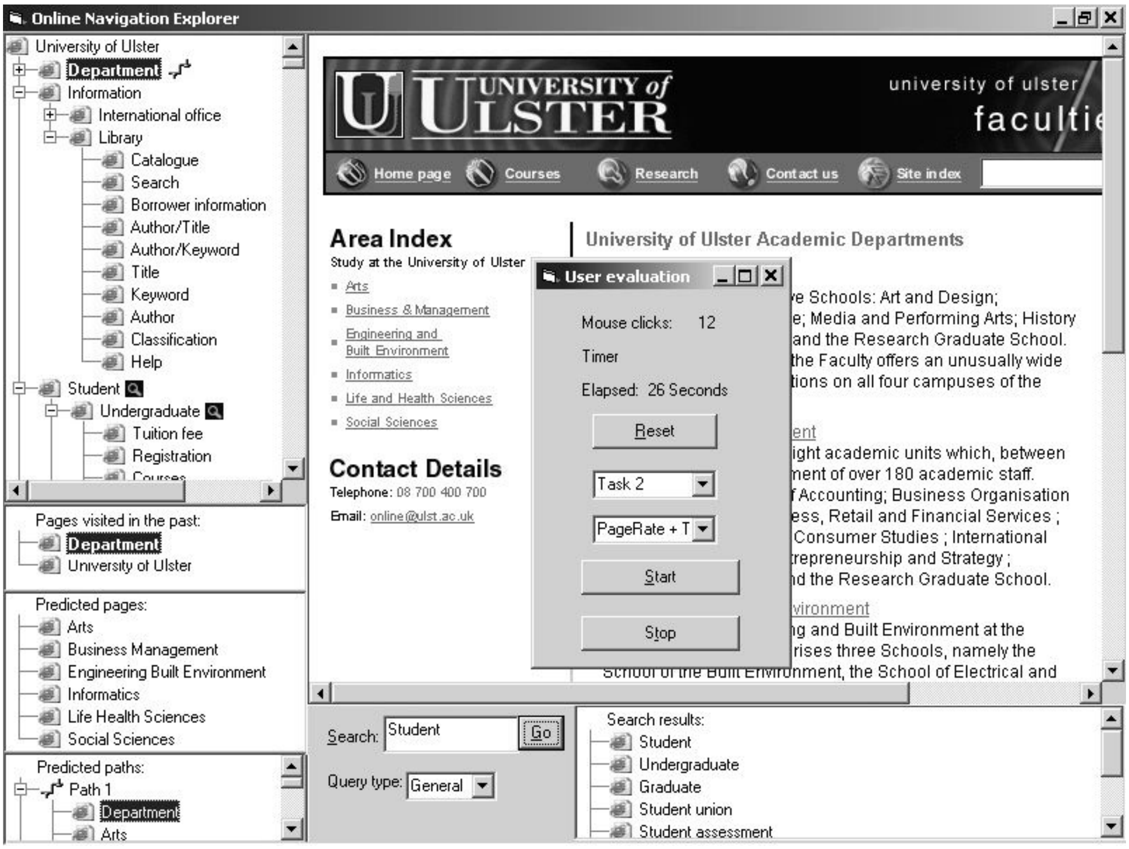
		Methods	Found/Target	Time (secs.)	Number of clicks
Task 1	Subject 1	Link prediction + tree view	1/1	31	2
		Tree view	1/1	35	2
		Link prediction	1/1	32	2
	Subject 2	Link prediction + tree view	1/1	31	2
		Tree view	1/1	30	2
		Link prediction	1/1	27	2
	Subject 3	Link prediction + tree view	1/1	23	2
		Tree view	1/1	24	2
		Link prediction	1/1	23	2
	Subject 4	Link prediction + tree view	1/1	27	2
		Tree view	1/1	33	2
		Link prediction	1/1	31	2
Task 2	Subject 1	Link prediction + tree view	2/2	33	5
		Tree view	2/2	42	6
		Link prediction	2/2	39	5
	Subject 2	Link prediction + tree view	2/2	53	7
		Tree view	2/2	63	7
		Link prediction	2/2	57	7
	Subject 3	Link prediction + tree view	2/2	69	6
		Tree view	2/2	75	6
		Link prediction	2/2	72	6
	Subject 4	Link prediction + tree view	2/2	42	4
		Tree view	2/2	47	4
		Link prediction	2/2	48	4
Task 3	Subject 1	Link prediction + tree view	9/12	309	20
		Tree view	9/12	343	23
		Link prediction	9/12	329	22
	Subject 2	Link prediction + tree view	10/12	317	23
		Tree view	9/12	373	27
		Link prediction	10/12	369	28
	Subject 3	Link prediction + tree view	11/12	421	18
		Tree view	11/12	472	22
		Link prediction	11/12	463	20
	Subject 4	Link prediction + tree view	10/12	389	19
		Tree view	9/12	408	20
		Link prediction	10/12	403	18

User feedback also suggested that link prediction integrated with the visualized conceptual link hierarchy is more useful than both the visualized conceptual link hierarchy alone and link prediction alone in helping them find the desired information on the Web site. Comparing Table 6.9 with Table 6.8, we find that users can find the desired information most efficiently and effectively when being assisted by link prediction integrated with the visualized conceptual link hierarchy.

6.4 Evaluation of Approaches to Ranking Search Results in Response to Users' Keyword-based Queries

In this section, we evaluate our approaches to ranking search results in response to users' keyword-based queries for adaptive Web site search. The goal of our experiments is twofold. First, compare the effectiveness and efficiency of search results ranked by the PageRate and PageRank algorithms in helping users search for desired information. Second, compare the effectiveness and efficiency of search results ranked by the PageRate algorithm on a Web site link structure and search results ranked by the PageRate algorithm on a link hierarchy integrated with the visualized link hierarchy in helping users search for the desired information.

In Figure 6.6, search results are listed in order of their overall rankings. Search results on a link hierarchy are integrated with the link hierarchy. The ico🔍 represents search results on the link hierarchy. Users can switch among the visualized link hierarchy, link prediction, and search to find the desired information.



Telephone: 08 700 400 700

Email: online@ulst.ac.uk

Mouse clicks: 12

Timer

Elapsed: 26 Seconds

Task 2

PageRate + T

Search: Student

Query type: General

ve Schools: Art and Design;
e; Media and Performing Arts; History
and the Research Graduate School.
he Faculty offers an unusually wide
ions on all four campuses of the

ent
ight academic units which, between
nent of over 180 academic staff.
f Accounting; Business Organisation
ess, Retail and Financial Services ;
Consumer Studies ; International
trepreneurship and Strategy ;
nd the Research Graduate School.

vironment
g and Built Environment at the
rises three Schools, namely the
School of the Built Environment, the School of Electrical and

Figure 6.6: Integrating search on a link hierarchy with a visualized link hierarchy in ONE.

Nine tasks of three types were given to 4 participants. Each type of tasks consists of three separate tasks of the same complexity. The three tasks are for search using the PageRank algorithm on a Web site link structure, search using the PageRate algorithm on a Web site link structure, and search using the PageRate algorithm on a link hierarchy integrated with the visualized link hierarchy, respectively. The three tasks of the first type are to find three particular persons' home pages, given their full names. The three tasks of the second type are to find the detail of research projects supervised by three lecturers, given their full names. Pages containing information about these research projects are either linked by their home pages or illustrated in their home pages, so to find their home pages is an important first step. The three tasks of the third type are to find papers published by three researchers in the last three years, given their full names. Some pages containing information about these papers are linked by their home pages. Co-authors of these papers may be mentioned in their home pages, which can be used for further search. Again to find their home pages is an important first step. As we can see in Table 6.10, most of the time, the participants performed better in search using the PageRate algorithm than search using the PageRank algorithm, and in search using the PageRate algorithm integrated with the visualized link hierarchy than search using the PageRate algorithm alone in terms of the success rate of finding targets as well as the time and the number of clicks used.

Table 6.10: The results of evaluating the effectiveness and efficiency of search using the PageRank algorithm, the PageRate algorithm, and the PageRate algorithm integrated with a visualized link hierarchy, respectively.

		Methods	Found/Target	Time (secs.)	Number of clicks
Task 1	Subject 1	PageRank	1/1	16	2
		PageRate	1/1	17	2
		PageRate + Tree view	1/1	15	2
	Subject 2	PageRank	1/1	17	2
		PageRate	1/1	16	2
		PageRate + Tree view	1/1	18	2
	Subject 3	PageRank	1/1	13	2
		PageRate	1/1	14	2
		PageRate + Tree view	1/1	14	2
	Subject 4	PageRank	1/1	12	2
		PageRate	1/1	11	2
		PageRate + Tree view	1/1	12	2
Task 2	Subject 1	PageRank	7/11	113	12
		PageRate	7/12	118	15
		PageRate + Tree view	9/13	93	12
	Subject 2	PageRank	7/12	123	18
		PageRate	7/11	111	14
		PageRate + Tree view	10/13	98	11
	Subject 3	PageRank	6/13	132	20
		PageRate	6/12	121	18
		PageRate + Tree view	8/11	103	14
	Subject 4	PageRank	8/11	102	16
		PageRate	9/13	107	18
		PageRate + Tree view	11/12	89	15
Task 3	Subject 1	PageRank	7/8	143	25
		PageRate	6/8	123	20
		PageRate + Tree view	8/9	103	17
	Subject 2	PageRank	6/8	136	21
		PageRate	8/9	130	20
		PageRate + Tree view	8/8	98	14
	Subject 3	PageRank	8/9	127	18
		PageRate	7/8	126	19
		PageRate + Tree view	8/8	102	13
	Subject 4	PageRank	6/8	148	27
		PageRate	6/8	139	26
		PageRate + Tree view	8/9	107	20

User feedback also suggested that search using the PageRate algorithm integrated with the visualized link hierarchy is more useful than search using the PageRate algorithm alone in helping them search for desired information on the Web site.

6.5 Summary

In this chapter, we have presented our experiments on the University of Ulster Web site to evaluate our approaches toward adaptive Web site navigation and search presented in chapter 3, 4, and 5, respectively. A prototype called ONE is used as the user interface for evaluation. First, we evaluated our approaches to mining link hierarchies and conceptual link hierarchies from Web log files. Our method can put pages onto the conceptual levels of a link hierarchy more accurately than both the breadth-first search method and the shortest weighted path method in link hierarchy construction. The PageCluster algorithm can cluster conceptually related pages more accurately than the bibliographic analysis method. The constructed conceptual link hierarchy is more compact than the link hierarchy. The visualized conceptual link hierarchy can help users find desired information more effectively and efficiently than browsing alone. Second, we evaluated our approaches to link prediction using Markov chain models. A transition matrix compression algorithm can compress the Markov chain models of Web site link structures (MMSs) for efficient link prediction. Link prediction using MMSs can help users find desired information more effectively and efficiently than browsing alone. Furthermore, link prediction using Markov chain models constructed from link hierarchies (MMHs) and conceptual link hierarchies (MMCs) integrated with visualized link hierarchies and conceptual link hierarchies can help users find desired information more effectively and efficiently than link prediction alone and visualized hierarchies alone. Third, we evaluated our approaches to ranking search results in response to users' keyword-based queries. Search using the PageRate algorithm can help users search for desired information more effectively and efficiently than search using the PageRank algorithm. Furthermore, by integrating search using the PageRate algorithm with visualized link hierarchies and link prediction, users can navigate and search for desired information effectively and efficiently.

Chapter 7

CONCLUSIONS

With the *ever-expanding* WWW, Web sites are getting increasingly complicated. It has been very difficult for users to find desired information on large Web sites. On the other hand, the link structure of a Web site consisting of information about Web site contents, hyperlinks, and user behavior can be mined for knowledge about the Web site and its users. The knowledge can be used to assist users to find desired information on the Web site *effectively* and *efficiently*. Since the two predominant paradigms for finding information on a Web site are*navigation* and *search*, we propose approaches to mining the Web site link structure for designing an adaptive Web site, which can automatically change its presentation and organization to assist user navigation and search.

7.1 Summary

We view Web site adaptation as a two-step process of first modeling users and then adapting the Web site to best meet the needs of each user or a group of users. In this thesis, we have described this process in detail in chapters 3, 4, and 5, built a prototype called ONE based on this architecture in chapter 6.

This thesis addresses two primary issues, namely, adaptive Web site navigation and adaptive Web site search. Nielsen [2000] identified three fundamental navigation questions that users might ask when they navigate a Web site, namely, Where am I now? Where have I been? Where can I go next? We address the first issue in two steps as follows.

First, in chapter 3, we visualize a Web site in a hierarchy in order to answer the first two navigation questions, namely, Where am I now? Where have I been? The hierarchy can be constructed based on the Web site link structure and how a group of users have used the Web site link structure. In the constructed hierarchy, the Web site link structure created by the designer is adapted to the views of users in using the Web site link structure, and can thus reflect users' views of the Web site. The hierarchy can help users understand the relationships between the pages they have visited and their current locations in the context of different levels of pages in the hierarchy. Using the hierarchy, they can control their navigation and thus are not confined to following only hyperlinks in each page. Users can decide where they can go next by understanding their current locations and relationships between the pages they have visited. The hierarchy can be seen as a form of adaptation of the Web site link structure to user behavior.

Second, in chapter 4, we presented approaches to predicting user navigation in order to answer the third navigation question, namely, Where can I go next? We developed a two-step process of modeling and prediction. We use a collaborative approach for doing this, which assumes that a user behaves in a similar way as other users. A user model can be built using data from a group of users, and it is then used to make predictions about a new user in the absence of information about him/her. When a user visits a Web site, a user model built with the collaborative approach will use its information regarding the habits of all past users to the site in order to predict the pages the user is most likely to request next [Zukerman et al. 1999]. Link prediction can be integrated with visualized hierarchy to answer all the three navigation questions.

To address the second primary issue, in chapter 5, we presented approaches to ranking Web pages given user behavior on the Web link structure in order to help users search for desired information. A collaborative approach has been taken, which assumes that a user behaves in a similar way as other users. Rankings of pages can be obtained using data from a group of users, and they are then used to rank search results for an individual user in the absence of information about the user. When a user uses a keyword-based query to search a Web site, rankings regarding the collective behavior of the group of users to the Web site are used to rank search results.

In chapter 6, we present a prototype called ONE that automatically adapts Web site navigation and search experience for users. The Web site link structure is visualized as a hierarchy. Link prediction using a user model is integrated with the hierarchy to assist user navigation. Search results are integrated with the hierarchy to assist user search. By integrating navigation with search and facilitating transition between them, users can locate desired information matching their *complex* information searching tasks effectively and efficiently.

7.2 Contributions

This thesis has made four contributions as follows.

1. **Adapting Web site navigation to user behavior.**User behavior in the form of traversals on hyperlinks is taken into account in designing an adaptive Web site for user navigation. First, in chapter 3, we propose a novel method for constructing a link hierarchy of a Web site, which is adapted to user behavior. Second, in chapter 3, we propose the PageCluster algorithm to cluster conceptually related pages on each conceptual level of a link hierarchy, where user behavior is taken into account in measuring the conceptual similarity between pages. Third, in chapters 3 and 5, user behavior is taken into account in synthesizing titles for pages and clusters. Fourth, in chapter 4, we propose to construct Markov chain models from Web site link structures (MMSs), link hierarchies (MMHs), and conceptual link hierarchies (MMCs) for link prediction, respectively, where user behavior is taken into account in estimating transition probabilities between states of the Markov chain models.

2. **Adapting Web site search to user behavior.** User behavior in the form of traversals on hyperlinks is taken into account in ranking Web pages in response to users' keyword-based queries. In chapter 5, we propose the PageRate algorithm, which takes into account information about both hyperlinks and user behavior in ranking Web pages.

3. **Improvement of link prediction toward adaptive Web site navigation**First, a compression algorithm is used to compress the transition matrix of a Markov chain

model constructed from a Web site link structure for efficient link prediction. Second, we apply a maximal forward path method to user sequences to improve the accuracy of link prediction. Third, we propose to predict the most probably-to-be visited pages and clusters with the next$n$ steps, which could save users' time and clicks in finding desired information. Fourth, we propose to predict guided paths using a Markov chain model constructed from a link hierarchy or conceptual link hierarchy, which help users find a path through the hierarchy of a Web site in order to find desired information.

4. **A unified framework for adaptive Web site navigation and search**Inspired by Perkowitz and Etzioni's earlier work on adaptive Web sites [1997 and 1998], we develop a richer model of a Web site for user navigation and search. We propose to integrate visualized hierarchies with link prediction, and navigation with search to allow users to find answers to complex information seeking tasks. A prototype called ONE is proposed to evaluate our approaches presented in chapters 3, 4, and 5.

7.3 Future Work

No piece of work is ever truly finished. We now describe two directions for future work.

1. **Improvements of the study of user behavior in our approaches**User behavior in the form of traversals on hyperlinks is taken into account in our approaches in chapter 3, 4, and 5. User behavior may change over time and differ between different groups of users. We plan to study how to segment a Web log file into different periods of time to reflect change over time, and how to segment a Web log file into groups of users having similar navigation and search behavior.

   User behavior has effects on our approaches as follows. First, in chapter 3, we can construct different link hierarchies of a Web site. The PageCluster algorithm clusters pages on each conceptual level of the link hierarchy differently. The synthesized titles of pages and clusters change. Second, in chapter 4, Markov chain models constructed from Web site link structures (MMSs), link hierarchies (MMHs), and conceptual link hierarchies (MMCs) are different. The compression algorithm compresses the transition matrix of a MMS differently. Thus different link prediction results are generated using these Markov chain models given a user sequence. Third, in chapter 5, the PageRate algorithm gives different authority-based rankings to pages. Relevance-based rankings of pages change since user behavior is taken into account in synthesizing feature vectors for pages.

We intend to study the relationships between user behavior and the changes in the link hierarchy, conceptual link hierarchy, Markov chain models, link prediction, and rankings of search results. By comparing these changes over time and across groups, we can investigate some unknown user behavior changes. This study also helps answer the question of how to segment a Web log file so that user behavior can be reflected in the link hierarchy, conceptual link hierarchy, link prediction, and search for user navigation and search.

2. **Improvements of ONE for adaptive Web site navigation and search.**By combining the functionalities of search and navigation into a single interface, ONE can help users accomplish complex information seeking tasks. As future work we plan to evaluate the effectiveness and efficiency of our approaches on other Web sites and with other information seeking tasks. We plan to test the scalability of ONE to a large volume of user requests, since users are treated independently in our approaches. We plan to investigate and compare different techniques for visualizing and integrating hierarchies, link prediction results, and search results for user navigation and search.

7.4 Final Words

The World Wide Web is a relatively new phenomenon in our history, but it is growing rapidly and has become the largest information repository in the world. WWW has penetrated almost everywhere in our lives and it has become a necessity to make it *accessible*, *understandable*, and *adaptable* to the needs of people. The Web can be seen as a huge social network created by the collaborative efforts of millions of Web site designers, Web content authors, and Web users. Despite the semi-structured, complex, massive, and heterogeneous nature of the Web,*human factors* in the Web can be explored for interesting knowledge about the Web and its users. The knowledge in turn can be used to help make the Web more usable for users. We have described one such exploration of making Web sites easier to use by adapting them to the way people use them.

The Web domain has certain characteristics that make it a fertile field for research:

- The availability of massive amount of information about Web contents, hyperlinks, and user behavior.

- A huge number of Web users who serve as both subjects for research and judges of the improvements made by research.

- Research of the Web can benefit a lot from various fields such as artificial intelligence, data mining, and machine learning.

As the Web flourishes, so does research of making it more usable for people.

BIBLIOGRAPHY:

Almind, T. C. and Ingwersen, P. (1997) Informetric Analysis on the World Wide Web: Methodological Approaches to “Webometrics”. *Journal of Documentation* 53, no. 4: 404-426.

Anderson, C. R., Domingos, P., and Weld, D. S. (2001) Adaptive Web navigation for wireless devices. In *Proc. of IJCAI’01*, pp. 879-884.

Anderson, C. R. (2002) *A Machine Learning Approach to Web Personalization* Ph.D. thesis. University of Washington, Department of Computer Science and Engineering.

Anderson, C. R., Domingos, P., and Weld, D. S. (2002) Relational Markov models and their application to adaptive Web navigation. In *Proc. of ACM SIGKDD’02*, Edmonton, Alberta, Canada, pp. 143-152.

Baeza-Yates, R. A. (1992) *Introduction to data structures and algorithms related to information retrieval* Information retrieval: data structures and algorithms, Prentice-Hall, Inc., Upper Saddle River, NJ.

Balabanovic, M., and Shoham, Y. (1997) Fab: Content-Based, Collaborative Recommendation. *ACM CACM* 40(3):66-72.

Banejree, A. and Ghosh, J. (2001) Clickstream clustering using weighted longest common subsequences, In *Proc. of Workshop on Web Mining* April, Chicago, Illinois, USA.

Bestavros, A. (1995) Using speculation to reduce server load and service time on the WWW. In *Proc. of 4th ACM International. Conference on Information and Knowledge Management (CIKM)* ACM, pp. 403-410.

Bharat, K. and Broder, A. Z. (1998) A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines. *WWW7 / Computer Networks* 30(1-7): 379-388.

Bharat, K. and Henzinger, M. R. (1998) Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. of 21st International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 104-111, August.

Bollen, J. and Heylighen, F. (1998) A system to restructure hypertext networks into valid user models. *The new review of Hypermedia and Multimedia*, 4:189-213.

Borges, J. (2000) *A Data Mining Model to Capture User Web Navigation Patterns*. PhD thesis. Department of Computer Science, University College London, London University.

Borges, J. and Levene, M. (1999) Data mining of user navigation patterns. In *Web Usage Analysis and User Profiling*, pp. 92-111. Published by Springer-Verlag LNCS, Vol. 1836.

Botafogo, R. A., and Shneiderman, B. (1991) Identifying Aggregates in Hypertext Structures. In *Proc. of ACM Hypertext'91*, pp.63-74.

Botafogo, R. A., Rivlin, E., and Shneiderman, B. (1992) Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. *ACM Transactions on Information System (TOIS)*, Vol. 10, No. 2, pp. 142-180

Botafogo, R. A. (1993) Cluster Analysis for Hypertext Systems. In *Proc. of ACM SIGIR'93*, pp.116-125.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. In *Proc. of WWW7*, pp. 107-117, Brisbane, Australia.

Buyukkokten, O., Kaljuvee, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. (2002) Efficient web browsing on handheld devices using page and form summarization. *ACM Transactions on Information Systems (TOIS)* 20(1): 82-115.

Cadez, I. V., Heckerman, D., Meek, C., Smyth, P., and White, S. (2003) Model-based clustering and visualization of navigation patterns on a Web site. *Journal of Data Mining and Knowledge Discovery*, in press.

Carpenter, M. P. and Narin, F. (1973) Clustering of Scientific Journals. *Journal of the American Society for Information Science*, 24(6), Nov.-Dec.

Chakrabarti, S., Dom, B. E., Gibson, D., Kleinberg, J. M., Raghavan, P., and Rajagopalan, S. (1998). Automatic resource list compilation by analyzing hyperlink structure and associated text. *WWW7 / Computer Networks* 30(1-7): 65-74.

Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., and Kleinberg, J. M. (1999) Mining the Web's link structure. *IEEE Computer*, 32(8), pp. 60-67.

Chakrabarti, S. (2002) *Mining the Web: Discovering Knowledge from Hypertext Data.* Morgan Kaufmann Publishers, ISBN: 1-55860-754-4.

Chen, C. (1998) Generalized similarity analysis and path finder network scaling. *Interacting with Computers*, 10(1998), pp. 107-128.

Chen, M. S., Park, J. S., and Yu, P. S. (1998) Efficient Data Mining for Path Traversal Patterns.*IEEE TKDE* 10(2): 209-221.

Chen, H. M., and Cooper, M. D. (2002) Stochastic modeling of usage patterns in a web-based information system. *Journal of the American Society for Information Science* 53(7): 536-548.

Conklin, J. (1987) Hypertext: An introduction and survey*IEEE computer,* Vol. 20, No. 9:125-156.

Cormen, T., Leiserson, C., and Rivest, R. (1990)*Introduction to Algorithms* MIT Press.

Crangle, C. E. (2002) Text Summarization in Data Mining. In*Proc. of Soft-Ware 2002: First International Conference on Computing in an Imperfect World,* pp. 332-347.

Crouch, D. B., Crouch, C. J., and Andreas, G. (1989) The use of cluster hierarchies in hypertext information retrieval. In*Proc. of Hypertext'89,*pp. 225-237.

Cunliffe, D., Taylor, C., and Tudhope, D. (1997) Query-based Navigation in Semantically Indexed Hypermedia. In *Proc. of Hypertext'97,* pp. 87-95, pp.245-246.

Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992) Scatter/Gather: A cluster based approach to browsing large document collections. In *Proc. of ACM SIGIR'92*, pp. 318-329.

Cutting, D. R., Karger, D. R., and Pedersen, J. O. (1993) Constant interaction-time scatter/gather browsing of very large document collections. In*Proc. of ACM SIGIR'93*, pp. 126-134.

Danilowicz, C., and Balinski, J. (2001) Document ranking based upon Markov chains. *Information Processing & Management* 37(4): 623-637, July.

Davison, B. D. (2000) Topical locality in the Web. In*Proc. of 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pp. 272-279, Athens, Greece, July.

Dean, J., and Henzinger, M. R. (1999) Finding Related Web Pages in the World Wide Web. In *Proc. of WWW8*, Elsevier Science, New York, pp. 389-401.

Dhyani, D., Ng, W. K. and Bhowmick, S. S. (2002) A Survey of Web Metrics. *ACM Computing Surveys*, Vol. 34, No. 4, December, pp. 469-503.

Diday, E. and Simon, J. C. (1976) *Clustering analysis. In Digital Pattern Recognition*, K. S. Fu, Ed. Springer-Verlag, Secaucus, NJ, pp. 47-94.

Ding, C. and Chi, C. H. (2000). Towards an adaptive and task-specific ranking mechanism in web searching. In *Proc. of ACM SIGIR'00*, pp. 375-376, Athens, Greece.

Ding, C., He, X., Husbands, P., Zha, H. and Simon, H. D. (2002) PageRank, HITS and a Unified Framework for Link Analysis. In *Proc. of 25th ACM SIGIR Conf.* pp.353-354, August, Tampere, Finland.

Douglas R. C., Jan O. P., David R. K., and John W. T. (1992) Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proc. of ACM SIGIR'92*, pp. 318-329.

Dumais, S. T. and Chen, H. (2000). Hierarchical classification of web content. In *Proc. of ACM SIGIR'00*, August, pp. 256-263.

Duran, B. S. and Odell, P. L. (1974) *Cluster Analysis: A Survey*. Springer-Verlag, New York, NY.

Durand, D. G. and Kahn, P. (1998) MAPA: A System for Inducing and Visualizing Hierarchy in Websites. In *Proc. of ACM Hypertext 1998*, pp. 66-76.

Dhyani, D. (2001) *Measuring the web: Metrics, models and methods*. Master's Dissertation. School of Computer Engineering, Nanyang Technological University, Singapore.

Eades, P., Lin, X. and Tamassia, R. (1996) An algorithm for drawing a hierarchical graph. *International Journal of Computational Geometry and Applications*, 6(2), pp. 145 - 155.

Etzioni, O. (1996) The World-Wide Web: Quagmire or Gold Mine? *ACM CACM* 39(11): 65-68.

Farkas, D. K. and Farkas, J. B. (2000) Guidelines for Designing Web Navigation. *Technical Communication*, 47(3), pp. 341-358, August.

Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. M. (2002) Self-Organization and Identification of Web Communities. *IEEE Computer*, Vol. 35 No. 3, pp. 66-71.

Fry, B. J. (2000) *Organic Information Design*. Master thesis, Massachusetts Institute of Technology, May.

Fu, Y., Sandhu, K., Shih, M.-Y. (1999) Clustering of Web users based on access patterns. In *Proc. of 1999 KDD Workshop on Web Mining*. Springer-Verlag, San Diego, CA, USA.

Fürnkranz, J. (1999) Exploiting Structural Information for Text Classification on the WWW. In D.J. Hand, J.N. Kok, and M.R. Berthold (eds.)*Advances in Intelligent Data Analysis: Proceedings of the 3rd Symposium (IDA-99)*pp. 487-497, Amsterdam, Netherlands, Springer-Verlag LNCS 1642.

Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998) CiteSeer: An automatic citation indexing system. In *Proc. of the third ACM conference on Digital Libraries*, pp. 89-98, Pittsburgh, PA, USA.

Glover, E. J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D. M., and Flake, G. W. (2002a) Using Web Structure for Classifying and Describing Web Pages. In *Proc. of WWW2002*, May 7-11, Honolulu, Hawaii, USA, ACM Press, pp. 562-569.

Glover, E., Pennock, D. M., Lawrence, S., and Krovetz, R. (2002b) Inferring Hierarchical Descriptions, In *Proc. of ACM CIKM'02*, November 4-9, McLean, Virginia, USA, pp. 507-514.

Gower, J. (1971) A General Coefficient of Similarity and Some of its Properties. *Biometrics* 27, pp. 857-874.

Guha, S., Bastogi, R., and Shim, K. (1998) CURE: An efficient clustering algorithm for large datasets, In *Proc. of SIGMOD'98*, pp. 73-84.

Hallam-Baker, P. M. and Behlendorf, B. (1996)*Extended Log File Format* W3C Working Draft WD-logfile-960323. http://www.w3.org/TR/WD-logfile.

Han, J. and Kamber, M. (2001)*Data Mining: Concepts and Techniques* The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor, Morgan Kaufmann Publishers, August. ISBN 1-55860-489-8.

Hand, D. J., Mannila, H., and Smyth, P. (2001)*Principles of data mining*, MIT Press.

He, X., Zha, H., Ding, C. and Simon, H.(2002) Web Document Clustering Using Hyperlink Structures. *Computational Statistics and Data Analysis*, Elsevier, Vol. 41, No. 1, pp. 19-45.

Hearst, M. A. and Pedersen, J. O. (1996) Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *Proc. of ACM SIGIR'96*, pp. 76-84.

Henzinger, M. R. (2000) Link Analysis in Web Information Retrieval*Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*Vol. 23, No. 3, pp. 3-8.

Henzinger, M, R. (2001) Hyperlink analysis for the Web*IEEE Internet Computing*, January/February, Vol. 5, No. 1, pp. 45-50.

Hodgson, J. (2001) Do HTML tags flag semantic content. *IEEE Internet Computing*. January/February, Vol. 5, No. 1, pp. 20-25.

Hong, J. (2000) Graph Construction and Analysis as a Paradigm for Plan Recognition. In *Proc. of AAAI-2000: Seventeenth National Conference on Artificial Intelligence*, AAAI Press/MIT Press, 30th July-3rd August, Austin, Texas, USA, 774-779.

Huberman, B. A., Pirolli, P. L., Pitkow, J. E., and Lukose, R. M. (1998) Strong Regularities in World-Wide Web Surfing. *Science* 280, no. 5360:94-97.

Jain, A. K. and Dubes, R. C. (1988) *Algorithms for clustering data.* Prentice Hall, 1988.

Jain, A. K., Murty, M. N. and Flynn, P. J. (1999) Data Clustering: A Review, *ACM Computing Surveys*, Vol. 31, No. 3, pp. 264-323, September.

Johnson, A. and Fotouki, F. (1994) Adaptive clustering of hypermedia documents, *Information Systems*, Vol. 19, No. 4, pp. 33-54.

Kaplan, C., Fenwick, J. and Chen, J. (1993) Adaptive Hypertext Navigation Based on User Goals and Context. *User Models and User Adapted Interaction*, 3(2), pp. 193-220.

Karypis, G., Han, E.-H. and Kumar, V. (1999) Chameleon: a hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, Vol. 32. No. 8, pp. 68-75, August.

Karypis, G. and Kumar, V. (1998) A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing* Vol. 20, No. 1, pp. 359-392, Society for Industrial and Applied Mathematics.

Kessler, M. M. (1963) Bibliographic Coupling Between Scientific Papers. *American Documentation*, 14(1):10-25, January.

King, B. (1967) Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86—101.

Kleinberg, J. M. (1998) Authoritative Sources in a Hyperlinked Environment. In *Proc. of Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM Press, New York, pp. 668-677.

Kleinberg, J. M. (1999) Authoritative Sources in a Hyperlinked Environment. *Journal of ACM* 46(5): 604-632.

Kleinberg, J. M. and Lawrence, S. (2001) The Structure of the Web. *Science* Vol. 294 30 November 2001, pp. 1849-1850.

Kleinberg, J. M. and Tomkins, A. (1999) Application of linear algebra in information retrieval and hypertext analysis. In *Proc. of 18th ACM Symp. Principles of Database Systems (PODS),* pp. 185-193, Philadelphia, PA, USA, May.

Kobsa, A. and Schreck, J. (2003) Privacy through Pseudonymity in User-Adaptive Systems. *ACM Transactions on Internet Technology (TOIT),* Vol. 3, No. 2, pp. 149-183, May.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R. and Riedl, J. (1997) GroupLens: Applying Collaborative Filtering to Usenet News. *ACM CACM* 40(3): 77-87.

Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999) Trawling the web for emerging cyber-communities. *WWW8/Computer Networks* 31(11-16): 1481-1493.

Larson, R. (1996) Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Proc. of the Annual Meeting of the American Society of Information Science,* Baltimore, Md., USA, Oct., pp.19 –24.

Lempel, R. and Moran, S. (2001). SALSA: the stochastic approach for link-structure analysis. *ACM Transations on Information Systems (TOIS),* 19(2):131--160, April.

Lieberman, H. (1995) Letzia: An Agent that Assists Web Browsing. In *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence,* Montreal, Canada, pp. 924--929.

Lovász, L. (1996) *Random Walks on Graphs: A Survey.* [in: Combinatorics, Paul Erdõs is Eighty, Vol. 2 (ed. D. Miklós, V. T. Sós, T. Szõnyi), János Bolyai Mathematical Society, Budapest, 1996, pp. 353—398.

McBryan, O. A. (1994) GENVL and WWWW: Tools for taming the web. In *Proc. of the First World-Wide Web Conference (WWW1),* 1994.

Michalski, R., Stepp, R. E. and Diday, E. (1981) A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts. In *Progress in Pattern Recognition,* Vol. 1, L. Kanal and A. Rosenfeld, Eds. North-Holland Publishing Co., Amsterdam, The Netherlands.

Minh, D. L. (2000) *Applied probability models,* Duxbury, Thomson Learning, ISBN 0-534-38157-X.

Modha, D. S. and Spangler, W. S. (2000) Clustering hypertext with applications to web searching. In *Proc. of ACM Hypertext'00,* pp. 143-152.

Motwani, R. and Raghavan, P. (1995) *Randomized Algorithms*. Cambridge University Press.

Mukherjea, S. and Hara, Y. (1997) Focus + Context Views of World-Wide Web Nodes. In *Proc. of ACM Hypertext'97*, pp. 187-196.

Munzner, T. (2000) *Interactive Visualization of Large Graphs and Networks*. Ph.D. Dissertation, Stanford University, June.

Nagy, G. (1968) State of the art in pattern recognition. In *Proc. of IEEE* 56, pp.836-862.

Ng, A. Y., Zheng, A. X. and Jordan, M.(2001) Link analysis, eigenvectors, and stability, In *Proc. of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, pp. 903-910.

Nielsen, J. (2000) *Designing Web Usability.* New Riders Publishing, Indianapolis, Indiana, USA.

Olston, C. and Chi, E. H. (2003) ScentTrails: Integrating Browsing and Searching on the Web. *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 10, No. 3, pp. 177-197.

Page, L., Brin, S., Motwani, R. and Winograd, T. (1998). The PageRank citation ranking: Bringing order to the Web. *Technical report*, Stanford University.

Padmanabhan, V. and Mogul, J. (1996) Using Predictive prefetching to improve World Wide Web latency. *Computer Communication Rev.* 26(3):22-36, July 1996.

Parunak, H. V. D. (1991) Ordering the information graph. In *Hypertext/hypermedia handbook*, ed. Emily Berk and Joseph Devlin. New York, NY: Intertext Publications, McGraw-Hill Publishing Co.

Perkowitz, M. and Etzioni, O. (1997) Adaptive web sites: an AI challenge. In *Proceedings of IJCAI'97*, pp.16-23.

Perkowitz, M. and Etzioni, O. (1998) Adaptive Web sites: automatically synthesizing Web pages. In *Proc. of AAAI'98*, pp. 727-732, July 26-30, Madison, Wisconsin, USA, ISBN 0-262-51098-7, AAAI Press.

Perkowitz, M. and Etzioni, O. (1999) Adaptive web sites: conceptual cluster mining. In *Proc. of IJCAI'99*, pp. 264-269.

Pirolli, P. and Pitkow, J. E. (1999) Distributions of Surfers' Paths Through the World Wide Web: Empirical Characterization. *World Wide Web* 1: 1-17.

Pitkow, J. and Pirolli, P. (1997) Life, death, and lawfulness on the electronic frontier. In *Proc. of ACM CHI'97*, pp. 383-390, Atlanta GA, USA.

Pitkow, J. and Pirolli, P. (1999) Mining longest repeating subsequences to predict world wide web surfing. In *Proc. of USITS'99: the 2nd USENIX Symposium on Internet Technologies & Systems*, Boulder, Colorado, USA.

Pitkow, J. E., Schütze, H., Cass, T. A., Cooley, R., Turnbull, D., Edmonds, A., Adar, E. and Breuel, T. M. (2002) Personalized search. *ACM CACM*, Vol. 45, No. 9, pp.50-55, September.

Rosenfeld, L. and Morville, P. (1998) *Information Architecture for the World Wide Web*. O'Reilly & Associates, Inc. ISBN: 1-56592-282-4.

Ross, S. (1983) *Stochastic Processes*. Wiley, New York.

Sarukkai, R. R. (2000) Link prediction and path analysis using Markov chains. *Computer Networks* 33(2000), pp. 377-386, Elsevier Science.

Silverstein, C., Henzinger, M. R., Marais, H. and Moricz, M. (1999) Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum* 33(1): 6-12.

Small, H. G. (1973). Co-Citation in the Scientific Literature: A New Measurement of the Relationship Between Two Documents. *Journal of the American Society of Information Science 24*, No. 4:265-269.

Small, H. G. and Koenig, M. E. D. (1977) Journal Clustering Using a Bibliographic Coupling Method. *Information Processing and Management*, 13(5):277-288.

Sneath, P. H. A. and Sokal, R. R. (1973) *Numerical Taxonomy*, San Francisco: W.H. Freeman, pp. 130-137.

Spears, W. M. (1998) A compression algorithm for probability transition matrices. *SIAM Matrix Analysis and Applications*, Vol. 20, No. 1, pp. 60-77.

Steinbach, M., Karypis, G. and Kumar, V.(2000) A Comparison of Document Clustering Techniques. In *Proc. of Text Mining Workshop*, ACM KDD.

Stewart, W. J. (1994) *An Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, NJ. 1994. ISBN 0-691-03699-3.

Wang, Y. and Kitsuregawa, M. (2001) Use link-based clustering to improve Web search results. In *Proc. of the 2nd International Conference on Web Information Systems Engineering (WISE'01)*, December 3-6, Kyoto, Japan, Vol. 1, pp. 115-124.

Weiss, R., Vélez, B. and Sheldon, M. A. (1996) HyPursuit: a hierarchical network search engine that exploits content-link hypertext clustering. In *Proc. of ACM Hypertext'96*, pp. 180 – 193, Bethesda, Maryland, United States.

Wexelblat, A. and Maes, P. (1997) Footprints: History-rich Web browsing. In *Proc. of Conf. Computer Assisted Information Retrieval (RIAO)*, pp. 75-84.

Willett, P. (1988) Recent trends in hierarchic document clustering: a critical review. *Information Processing and Management*, Vol. 24, No. 5, pp. 577-597.

Wishart, D. (2001) *Clustan Professional User Guide.* Clustan Ltd., Edinburgh, Scotland.

Wishart, D. (2002) k-means clustering with outlier deletion, for data mining with mixed variables and missing values. In: *Exploratory Data Analysis in Empirical Research*, Schwaiger, M, and Opitz, O (eds), Springer, pp.216-226.

Zamir, O. and Etzioni, O. (1999) Grouper: A Dynamic Clustering Interface to Web Search Results. *WWW8 / Computer Networks* 31(11-16): 1361-1374.

Yan, T., Jacobsen, M., Garcia-Molina, H. and Dayal, U. (1996) From user access patterns to dynamic hypertext linking. *WWW5 / Computer Networks* 28(7-11): 1007-1014.

Yang, Q., Zhang, H. and Li, T. (2001) Mining Web logs for predictions models in WWE caching and prefetching. In *Proc. of ACM SIGKDD'2001*, San Francisco, CA, USA, pp. 473-478.

Zhang, T., Ramakrishnan, R. and Linvy, M. (1996) Birch: an efficient data clustering method for large datasets. In *Proc. of SIGMOD'96*, pp. 103-114.

Zhu, J. (2001) Using Markov Chains for Structural Link Prediction in Adaptive Web Sites. In *Proc. of User Modeling 2001, 8th International Conference, UM 2001.* Lecture Notes in Computer Science 2109 Springer 2001, ISBN 3-540-42325-7: pp. 298-300, Sonthofen, Germany, July 13-17.

Zhu, J., Hong, J. and Hughes, J. G. (2001) PageRate: Counting Web users' votes. In *Proc. of the Twelfth ACM conference on Hypertext and Hypermedia (Hypertext'01)*, pp. 131-132, ACM Press, Århus, Denmark, August.

Zhu, J., Hong, J. and Hughes, J. G. (2002a) Using Markov Chains for Link Prediction in Adaptive Web Sites. In *Proc. of Soft-Ware 2002: First International Conference on Computing in an Imperfect World*, pp. 60-73, Lecture Notes in Computer Science, Springer, Belfast, April.

Zhu, J., Hong, J. and Hughes, J. G. (2002b) Using Markov Models for Web Site Link Prediction. In*Proc. of the Thirteenth ACM conference on Hypertext and Hypermedia (Hypertext'02),* pp. 169-170, ACM Press, College Park, MD, USA, June 11-15.

Zhu, J., Hong, J. and Hughes, J. G. (2003) PageCluster: Mining Conceptual Link Hierarchies from Web Log Files for Adaptive Web Site Navigation*ACM Transactions on Internet Technology (TOIT),*in press.

Zukerman, I., Albrecht, D. and Nicholson, A. (1999) Predicting Users' Requests on the WWW. In *Proc. of UM99 Proceedings – the Seventh International Conference on User Modeling* pp. 275-284, Banff, Canada, Springer-Verlag.