Self Adapting Websites: Mining User Access Logs

Ganesan Bathumalai

A thesis submitted in partial fulfilment
of the requirements of
The Robert Gordon University
for the degree of Master of Research

June, 2008

Abstract

The Web can be regarded as a large repository of diversified information in the form of millions of websites distributed across the globe. However, the ever increasing number of websites in the Web has made it extremely difficult for users to find the right information that satisfies their current needs. In order to address this problem, many researchers explored Web Mining as a way of developing intelligent websites, which could present the information available in a website in a more meaningful way by relating it to a user's need. Web Mining applies data mining techniques on web usage, web content or web structure data to discover useful knowledge such as topical relations between documents, users' access patterns and website usage statistics. This knowledge is then used to develop intelligent websites that can personalise the content of a website based on a user's preference. However, existing intelligent websites are too focussed on filtering the information available in a website to match a user's need, ignoring the true source of users' problems in the Web. The majority of problems faced by users in the Web today, can be reduced to issues related to a website's design. All too often, users' needs change rapidly but the websites remain static and existing intelligent websites such as customisation, personalisation and recommender systems only provide temporary solutions to this problem. An idea introduced to address this limitation is the development of adaptive websites. Adaptive websites are sites that automatically change their organisation and presentation based on users' access patterns. Shortcutting is a sophisticated method used to change the organisation of a website. It involves connecting two documents that were previously unlinked in a website by adding a new hyperlink between them based on correlations in users' visits. Existing methods tend to minimize the number of clicks required to find a target document by providing a shortcut between the initial and target documents in a user's navigational path. This approach assumes the sequence of intermediate documents appearing in the path is insignificant to a user's information need and bypasses them. In this work, we explore the idea of adaptive websites and present our approach to it using wayposts to address the above mentioned limitation. Wayposts are intermediate documents in a user's path which may contain information significant to a user's need that could lead him to his intended target document. Our work identifies such wayposts from frequently travelled users' paths and suggests them as potential navigational shortcuts, which could be used to improve a website's organisation.

Acknowledgments

First and foremost I would like to express my utmost gratitude to Dr. Stuart Watt who instigated me to take up research. I would have never ventured into this path if it wasn't for his support, advice and guidance. Your dedication and continuous support towards me will always be remembered. Secondly I would like to extend my sincere gratitude to Prof. Susan Craw and Dr. Frank Herrmann who agreed to supervise my research at the School of Computing. I am in debt to them for their excellent supervision throughout my research at the Computing Technology Centre (CTC). Their advice and continuous support allowed me to improve in many aspects of research and writing skill. With their guidance I was able to complete this thesis on time.

I would also like to thank my colleagues from the CTC, for all their assistance, support, interest and valuable suggestions. I am especially obliged to Sutanu Chakraborti for his guidance with MatLab and stimulating discussion on some research problems, Stella Asimwe, Rahman Mukras and Stuart Massie for their assistance with LaTex. Finally I would like to dedicate this work to my family members who stood by me throughout the time of my needs. Their unwavering love and support kept me going even when I was feeling down and helpless. And of course my friends whom have always shown their support for me in all my endeavours. A great man once said, "Science without religion is lame. Religion without science is blind" (Albert Einstein). Therefore I wish to take this opportunity to express my gratitude to the Almighty himself who, provided me with the spiritual guidance needed to accomplish the task of completing my research.

"*Om Sri Ganeshwaraya Namaha*"

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The World Wide Web or simply "Web" originally started as a research development for information sharing purposes. However, today it consists of a vast heterogeneous information source distributed over numerous websites all over the world. It has become a part of life for almost everyone around the globe who have grown to be dependent on this technology. For example communication via email, messaging services (e.g., chat groups), up-to-date news, blogging, online shopping and weather reports are a few elements of the Web that have influenced many people's lives. In addition, more and more business organizations are making their presence on the Web to increase their business flexibility and earning potential to be able to compete on a more level playing field with other e-commerce businesses. The Web, officially founded in 1989 by Sir Tim Berners-Lee, consisted of no more than a single web server (i.e., website) with a single page describing the project itself. Since then the growth of the Web continues to be exponential and it has not ceased to grow. In an article released by BBC (2005) it was reported that the Web had a staggering 74.4 million registered websites online in October 2005. On the other hand, a recent survey conducted by Netcraft (2008) in April 2008 noted that the Web had 165.7 million websites online with the month's most growth of 3.1 million sites coming from the United States and 1.1 million sites of these were from Google's blogger service. It can be seen that the Web has grown more than twice its size within just a span of $1\frac{1}{2}$ years between October 2005 and April 2008. This suggests that the Web will continue to grow as long as there are people who keep developing it.

However, this rapid information boom in the Web is not without consequences. The fact that there are millions of websites, no official governance and more websites emerging every day has significantly affected the ability of users to find the right information that satisfies their needs. Furthermore, many websites are static in nature and only serve the purpose of relaying information. Static websites are cheap and easy to create but the information contained in these websites is fixed upon publishing and cannot be changed until the developer decides to publish a newer version of the website. Updating a static website requires professional skill and this increases the cost of maintenance in the long run. Static websites are developed to cater for the needs of an organization or its developer rather than the users' need. Furthermore, different users have different perspectives of the data available in a website. Therefore, it is very likely that users will have diverse preferences on the type of information that they are looking for in a website. This introduces the need for intelligent websites which could assist users in finding the information that satisfies their on-line needs. As a result, many researchers turned to Web Mining as the means of developing intelligent websites which could cater for the different needs of the users.

Web Mining is the application of data mining techniques on web resources to find patterns in the Web. There are three kinds of web resources available, namely usage data, content data and structure data. As such, Web Mining can be distinguished into web usage mining, web content mining and web structure mining and combinations of them. A common source of usage data is web server logs otherwise known as web logs. Web logs are textual data collected by servers over a period of time, which are enriched with information on users' browsing behaviour and website usage statistics. It is a massive repository of users' activities in a website tracked by the web server. Web usage mining applies data mining techniques on this repository to discover useful knowledge about users' behaviour in a website. This knowledge can then be used for various web applications such as assisting a user to navigate the website and suggesting popular links to users based on their interest.

Web content data consists of textual information contained in the documents of a website. A document's textual information is highly unstructured and varies from one website to another. Unlike web logs which have a fixed set of information fields that can be used to interpret the data, no such precise reference frame is available to represent a document's content. Web content mining involves applying data mining techniques on the textual content of a document to discover knowledge such as topical relations between documents in a website. The knowledge can be used to create a topical hierarchy of the documents in a website and identify documents that match a user's need.

Web structure data refers to the hyperlinks interconnecting the collection of documents in a website. Each document in a website may have a number of outgoing and incoming hyperlinks. A link between two documents in a website suggests that the documents may be related and may contain relevant information. Web structure mining applies data mining techniques on this network of hyperlinks structure to extract information that can be used for a variety of purposes, for example web crawling.

The earliest approach to intelligent websites was customisation, which can be traced back to the mid 1990s. Customisation involves changing the interface and contents of a website to match a user's need. This is done through a manual form filling method which requires the users to choose from a set of predefined interest categories or specify their interests. Yahoo is one such website that allows users to customise the content presented to them by choosing from a predefined set of interest categories. The drawback of customisation is that it is time consuming to set up a complex customisation feature by filling forms online and users need to put effort in order to keep their profiles up-to-date, otherwise they will remain static. Website personalisation addresses this limitation through the means of automated user profiling. Automated user profiling refers to the use of implicit user profiling methods to procure a user's preference in a website. Personalisation uses web usage mining techniques to automatically create a user's profile from web logs and change the information in a document to suit the user. Following personalisation, many other advances in Web technology came about one such advance was recommender systems. Recommender systems are web applications that employ web usage mining techniques on web logs to come up with personalised recommendations on interesting information in a website. These recommendations could be as simple as suggesting popular links in a website or involve a more sophisticated real time approach, for example proposing links to documents that are related to a user's need as the user browses a website. All of the above mentioned intelligent web applications are primarily focussed on filtering the information available in a website to suit a user's need. It is obvious that the websites themselves do not change and as time passes, these websites are no more than "fossils cast in HTML" as described in (Etzioni and Perkowitz, 2000).

The majority of the problems faced by Web users these days such as the inability to find the documents they are searching for and difficulty in navigating a website, are caused by either a poor website design or the website being used in an unanticipated way that violates its original design purpose. In (Etzioni and Perkowitz, 1997), the authors challenged the AI community to address this problem by developing adaptive websites. They defined adaptive websites as sites that automatically improve their organisation and presentation by learning from visitors access patterns. This initiated the beginning of the development of adaptive websites that focus on modelling the website based on users' need rather than intelligent websites which model the users. The key to creating such adaptive websites is the transformation approach which involves in making changes to the basics of a website design using the information obtained from web logs. Since its introduction, a few studies both theoretical and practical has come about but the overall development of adaptive websites is still in its infancy.

1.1 Motivation for Work

The motivation for our work comes primarily from three separate studies into Web technology using Web Mining techniques. Each of these studies has contributed to different advancements in Web technology both practically and theorectically. The first is Wexelblat and Maes's (1999) study which was based on the metaphor of "*footprints*" left by travellers creating footpaths in the grass over time. In a website, visitors leave footprints behind indicating how frequently a link is traversed. These web trails (i.e., navigational paths) accumulated over time, reveal the most heavily travelled area of the website, and can be used to guide new visitors to the website as an indicator to a topic of interest worth visiting. This gave us our initial inspiration to explore the area of Web Mining and investigate the means of mining web logs for collective users' access patterns which could be used to improve a user's browsing experience in a website.

The second one is Etzioni and Perkowitz's (1997) study which introduced the idea of adaptive websites: sites that are capable of changing their presentation and organisation based on users' access patterns. In this study, the authors pointed out that users' needs and their browsing patterns are constantly changing over time while the websites remain static. In order to address this they put forth a challenge to the AI community to create adaptive websites. Since then, a few studies have been undertaken to develop adaptive websites, hence it also inspired us to look into the possibilities of an adaptive website. Furthermore the current advancements in Web technology such as customisation, personalisation and recommender systems only filter the information in a website to match the users' need. If one observes closely it is evident that the problems such as finding a specific document of interest and difficulty in navigating a website stem directly from website structural design related issues. The advancements mentioned above only provide temporary solutions to these design problems rather than addressing it. This further instigated us to explore the idea of adaptive websites and present an approach which could be used to transform a website's design based on users' access patterns.

Finally Lee and Shiu's (2004) study provided us with the direction of our own research problem. Their study introduced an adaptive website that makes changes to its organisation by adding and removing hyperlinks between documents. The authors claimed that longer traversal paths lead to higher chances of users making mistakes while browsing for specific documents of interest. They suggested that a website's efficiency can be improved by suggesting shortcuts that reduce the length of the path a user needs to traverse to find a document. In this study, the authors did not make a clear explanation of how two documents will be selected for shortcutting. This factor influenced us to investigate existing studies that employ shortcutting to improve a website's organisation. The result of the investigation is the introduction of our approach to adaptive websites using wayposts. In the investigation, we found existing studies usually provide shortcuts by creating a link between the initial and final document of a user's traversal path, disregarding the intermediate documents the user visited (Srikant and Yang, 2001; Lee and Shiu, 2004; Brickell, Dhillon and Modha, 2007). However, we believe that these intermediate documents may contain information that is of significance to a user's need and could be used as wayposts (i.e., navigational shortcuts) to aid him in reaching his document of interest.

1.2 Research Objectives

This thesis reports on our approach to adaptive websites. With respect to this, it explores web usage mining techniques that could be applied to web logs to discover users' paths and identify wayposts from them which, can be used to suggest potential shortcuts between documents in a website. Listed below are a set of specific objectives that we will adddress:

1. *Identify a group of target documents based on users' access patterns and extract navigational paths terminating at these documents from users' sessions.*
2. *Group similar navigational paths together and find hyperlinks (i.e., wayposts) that can act as potential shortcuts points by identifying documents that commonly co-occur in these paths.*
3. *Evaluate the shortcuts generated from the wayposts on their usefulness in providing shortened navigational paths.*

1.3 Thesis Overview

In this chapter, we have highlighted the problems faced by users in the Web and the evolution in Web technology which serves as the motivation for the work reported in this thesis. Furthermore we have also outlined the specific objectives of our research.

Chapter 2 provides an overview of related works in the field of Web Mining with an emphasis on usage mining. It also briefly introduces the other two forms of web mining which are content and structure mining. The chapter also elaborates on the advancements of Web technology and provides an overview of related studies in developing intelligent websites which sets the backdrop for our work. Following that, this chapter presents the idea of adaptive websites and describes in detail the transformation approach on which it is based. Then we introduce two methods in which a website could be transformed, namely organisation adaptation and presentation adaptation. Finally, we present studies that employ these methods to create adaptive websites and identify their limitations which help to position our work in its context in the following chapters.

Chapter 3 introduces our approach to adaptive websites and discusses some of the tasks carried out with respect to the objectives outlined. We specifically look at the process of acquiring data from raw web logs that consists of the following 3 tasks: web logs filtration, crawler logs elimination and sessionisation. The nature of these tasks and the methods employed to achieve them are discussed in detail.

Chapter 4 presents the tasks that were carried out with respect to extracting knowledge from the data acquired of web logs. These tasks include identifying target document in a website and discovering paths from them, grouping similar paths together and finally identifying wayposts for shortcuts generation. Each tasks and the method employed to achieve them are elaborated in detail. Finally an evaluation of our approach to adaptive websites is presented.

Finally chapter 5 summarises the contributions of this research, identifies some of its limitations and suggests possible extensions for future research.

Chapter 2

Literature Review

This chapter aims to provide a detailed review of related works in the field of Web Mining and Adaptive Websites. In particular, we will focus on the type of data used in usage mining, challenges to usage mining and the various forms of its application in the Web. Apart from that, we will also briefly address the other two forms of Web Mining namely content and structure mining. However, note that usage, content and structure mining constitutes broad Web Mining topics on their own. Therefore we will cover as much ground necessary to set the backdrop for our research rather than attempting a comprehensive review of the three. Finally, we will look at the advancements of websites from a collection of static documents to interactive web applications known as adaptive websites.

The chapter is organised as follows: In the following section, the fundamentals of web usage mining is explained, followed by a discussion on issues related to mining web logs and applications of usage mining to the Web. Section 2.2 introduces the two other forms of Web Mining and their applications to the Web. It also highlights how recent studies have began using combinations of the three forms of Web Mining. Section 2.3 reviews of the development of websites technologies with respect to the advancements in the Web Mining field. Each subsection in section 2.3.2 explores existing approaches to adaptive websites and identifies their limitation which serves as motivation for our research. Finally, section 2.5 concludes the chapter.

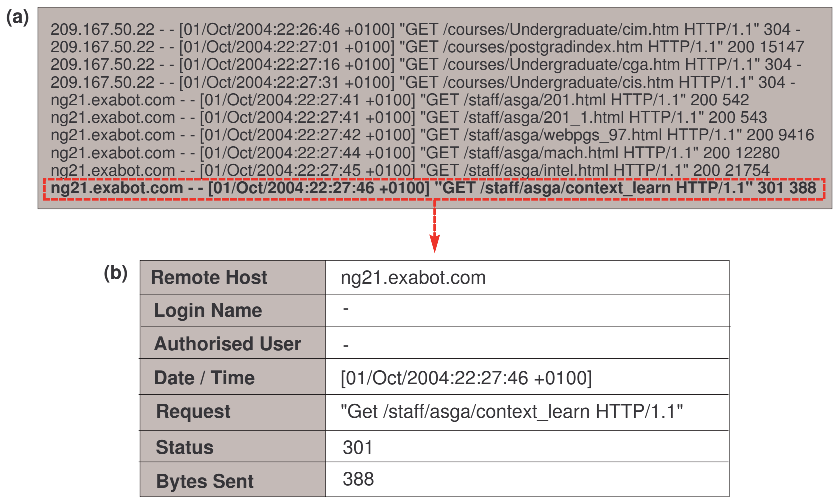
2.1 Web Usage Mining

According to a recent survey, the number of web usage mining studies undertaken over the past few years has been rapidly increasing (Facca and Lanzi, 2005). Web usage mining is the application of data mining techniques on large web repositories to discover useful knowledge about users behavioural patterns and website usage statistics that can be used for various website design tasks. The main source of data for web usage mining consists of textual logs collected by numerous web servers all around the world. This is probably because web logs are the easiest and cheapest way of gathering information about the users and a website. Other sources of usage data may include proxy servers and client side logs.

Web logs (i.e., web server logs) are also commonly referred to as user access logs, user trails and click-stream data by the Web Mining community. Web logs are trails of past activities left behind by users of a website. These historical logs are embedded with significant information about the users and how a website is being used on a day-to-day basis. Such information are deemed invaluable in today's world of customer-oriented businesses, especially for companies that rely on the web to advertise their services (e-commerce) and to web designers who wish to maintain a constant stream of visitors to their website. Web usage mining provides these entities with the means of discovering such information that can be used to improve a business's performance or increase the effectiveness of a particular website.

2.1.1 Types of Web Logs

Web logs come in various formats but most logs appear in the Common Log Format (CLF). The CLF is the simplest and standard type of web logs format that is widely used by most web servers. This is because CLF is supported by the majority of the commercially available analysis tools, which could be used to extract information on website usage. Figure 2.1a shows an extract of web logs in CLF from the School of Computing of the Robert Gordon University. Each line in figure 2.1a refers to a document request made by a user to the server during his visit to the website. A document request (i.e., a CLF log entry) consists of a fixed number of information fields such as the address of the remote host, the date and time of the request and the URL requested among others. Figure 2.1b shows the information fields that make up a single document request and the information that may be obtained from each field.



(a)

```
209.167.50.22 - - [01/Oct/2004:22:26:46 +0100] "GET /courses/Undergraduate/cim.htm HTTP/1.1" 304 -
209.167.50.22 - - [01/Oct/2004:22:27:01 +0100] "GET /courses/postgradindex.htm HTTP/1.1" 200 15147
209.167.50.22 - - [01/Oct/2004:22:27:16 +0100] "GET /courses/Undergraduate/cga.htm HTTP/1.1" 304 -
209.167.50.22 - - [01/Oct/2004:22:27:31 +0100] "GET /courses/Undergraduate/cis.htm HTTP/1.1" 304 -
ng21.exabot.com - - [01/Oct/2004:22:27:41 +0100] "GET /staff/asga/201.html HTTP/1.1" 200 542
ng21.exabot.com - - [01/Oct/2004:22:27:41 +0100] "GET /staff/asga/201_1.html HTTP/1.1" 200 543
ng21.exabot.com - - [01/Oct/2004:22:27:42 +0100] "GET /staff/asga/webpgs_97.html HTTP/1.1" 200 9416
ng21.exabot.com - - [01/Oct/2004:22:27:44 +0100] "GET /staff/asga/mach.html HTTP/1.1" 200 12280
ng21.exabot.com - - [01/Oct/2004:22:27:45 +0100] "GET /staff/asga/intel.html HTTP/1.1" 200 21754
ng21.exabot.com - - [01/Oct/2004:22:27:46 +0100] "GET /staff/asga/context_learn HTTP/1.1" 301 388
```

(b)

Remote Host	ng21.exabot.com
Login Name	-
Authorised User	-
Date / Time	[01/Oct/2004:22:27:46 +0100]
Request	"Get /staff/asga/context_learn HTTP/1.1"
Status	301
Bytes Sent	388

Figure 2.1: (a) Common Log Format (b) Data Fields That Make Up A Single Request

The remote host field contains the address from which the request for the document originated. As it can be seen from figure 2.1a this field caters for both the canonical i.e., Domain Name Server (DNS), and numerical i.e., Internet Protocol (IP) form of the remote host address. This is because the value returned for this field is dependent on the client machine's settings which vary from one machine to another. The login name and authorised user fields are part of an authentication process with which the user on the other end of the remote host has authenticated himself with. It is a common practice among secured websites carrying sensitive information which, requires their users to authenticate themselves before being able to access the content of the website. These two fields, as their names suggest, contain the user name and password of a user requesting access to a particular document. They are usually replaced with a "-" symbol if such information is not available or applicable to the website[1].

The date and time field records the date and the time of the request in the following

[1]However note that, the values for these fields could be anonymised even if they are available due to security reasons by web servers

format: (Day/Month/Year : Time in 24-hours : Geographical Time Difference). Each document request made to the web server is organised according to the date and time it was made. The request field contains the request line as it came from the user and is composed of three parts. The first part describes the method used to obtain the document (e.g., GET, POST), the second part records the filename of the document that is to be fetched and finally the third part indicates the version of the Hypertext Transfer Protocol (HTTP) that was used. Following that the status field records the HTTP response code returned from the server indicating whether the file was successfully retrieved or not. The status field contains an integer ranging from 100 to 500, which represents the different response codes that may be received from the server given a user's request. A status code of 200 indicates that the document requested was successfully retrieved[2]. The final field, bytes sent, simply refers to the number of bytes transferred back to the client which is the size of the file retrieved from the web server.

Another format in which the web logs may appear is the Extended Log Format (ELF). The ELF was mainly introduced to address the limitation to the amount of information that can be extracted from CLF web logs. As mentioned previously, each of the CLF logs carry a fixed number of data fields that contain information about a user and his document request. However, in many cases it is desirable to record more information about the user and his activities in the website (e.g., time spent on a document). The main property of ELF logs is its extensibility which allows one to dictate what data needs to be recorded, thus permitting a wider range of data capture. However, compared to CLF, the ELF is not widely used with only a handful of web servers implementing it. A full description of the ELF, complete with an example of ELF logs, could be found at (http://www.w3.org/TR/WD-logfile.html).

We will be working extensively with CLF-based web logs throughout this thesis, as our main source of data comes from the Robert Gordon University's School of Computing web server which employs CLF.

[2]The response codes are divided into five main classes and a detailed description of each class can be found in appendix B or alternatively at http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

2.1.2 Challenges in Usage Mining

There are two main challenges to mining web logs, which need to be addressed in order to discover useful knowledge from them. The first challenge is the volume of document requests that are amassed in a single log file. The log file can be regarded as a huge collection of records of events that took place in a website. It includes entries of document traversal, file retrieval and unsuccessful web events among many others that are organised according to the date and time it was made to the web server. This sheer volume and diversity of records makes it necessary to distinguish between document requests that carry significance in inferring users behaviour and website usage information over those that do not. Note that distinguishing which document requests is needed and which is not, may vary significantly based on the purpose and specification of the study undertaken. For example, requests for graphical content are generally considered to be insignificant and filtered out. However, this may not be desirable in the case of a website dedicated to photography, where most of its content would be graphical in nature. Thus, it is important to carefully consider which type of document requests will maximise the chances of discovering knowledge about users and the website. Otherwise it could result in the removal of document requests that carry vital information, thus leading to data loss and undesirable effects of the study undertaken.

The second challenge involves finding and grouping all document requests that was made by the same user during the user's visit to the website. This is commonly referred to as session modelling or otherwise sessionisation. This task can be difficult and is highly dependent on the type of information available in the web logs. The term 'session' is generic and could mean different things to different people based on the scope and purpose of one's study. Cooley, Mobasher and Srivastava (1999) defined a session to be all of the document accesses that occur during a single visit by a user to a website. On the other hand, Etzioni and Perkowitz (2000) defined a user's session as a series of document views in a day's log from one visitor. Note how the former definition fails to mention what is meant by a single visit while the latter specifically states the period of a single visit to be one whole day. This suggests that a user could make multiple number of visits to a website in a day and that a single visit to a website may also consists of one or more number of sessions. There are two major issues in identifying a user's session, namely local caching and proxy servers.

Local caching is carried out by web browsers in order to improve performance and minimise network traffic. For every request made by a client the browser saves a copy of the requested document on the client's machine. This results in the cached document to be displayed whenever a user clicks the 'back' button in the browser thus making the web server unaware of the repeated document request. Reconstructing users' sessions from web logs with such missing data may lead to inconsistency in the navigational routes found. There are techniques available to overcome this problem such as the use of referrer log (Cooley et al., 1999) and modified browsers (Catledge and Pitkow, 1995). These techniques provide detailed information about a user's browsing behaviour and allow the system to track the sequence of document requests made by the user. However they are heavily dependent on users' co-operation and are restricted by privacy issues.

Many Internet Service Providers and large companies provide improved navigation and download speed to users by employing proxy caching. This is referred to as proxy server services and creates even more problems in identifying a user's session. All document requests made through a proxy server have the same originating DNS/IP address even though the requests are likely to represent multiple users. This makes its extremely difficult to use the remote host field (originating DNS/IP address) of a web log as an identifier to distinguish between requests made by different users. Proxy caching problems could be solved by the use of cookies or URL rewriting (Mobasher, Cooley and Srivastava, 2000) to include a session identifier which can then be used to identify users' sessions. However, not all web browsers support cookies while URL rewriting yet again raises issues of privacy. Consequently (Ansari, Kohavi, Mason and Zheng, 2001) suggested directly accessing the server application layer to overcome proxy caching. However, this method is not widely applicable due to server application copyright issues and the need to customise the web mining application to a specific server. Nevertheless, there are other more sophisticated solutions available to identifying users' sessions by mean of heuristic such as those introduced in (Pirolli, Pitkow and Rao, 1996; Pitkow, 1997; Berendt, Mobasher, Nakagawa and Spiliopoulou, 2002).

2.1.3 Applications of Usage Mining

The fact that web logs are the easiest and cheapest way of gathering information about the users and a website has been a catalyst in the development of various web applications employing usage mining techniques. There are various web usage mining techniques and algorithms currently available but the most commonly used ones are association rule mining, sequential pattern learning and clustering. Association rule mining consists of implication rules which are used to find associations among documents appearing together in users' visits. A typical example of a rule may be: "*55% of visitors who accessed library.hmtl and catalogue.html also accessed account.html*". Sequential pattern learning is similar to association rule mining but is used to find a temporal relationship between documents of a website. It gives importance to the order in which the documents are accessed by a user. Clustering is used to group a set of items that share similar characteristic together. A more detailed overview of the above algorithms and their applications can be found in (Cooley et al., 1999; Srivastava, Cooley, Deshpande and Tan, 2000).

The resulting knowledge from mining web logs can be used for a variety of tasks such as web design, web browsing, web searching and web crawling. For example, Cooley et al.'s (1999) WEBMINER system uses association rule mining and sequential pattern learning to extract useful statistics about documents that are frequently accessed together by users. WEBMINER pre-processes the web logs into a series of transactions and analyses them using association rule mining and sequential pattern learning to discover documents that frequently co-occur in users' visits and the duration spend on these documents. This information could then be used to improve the website design by adding link between frequently co-occurring unlinked documents. Similarly Masseglia, Poncelet and Teisseire (2003) introduced WebTool, an application that uses association rule mining technique to analyse web logs and identify relations among visited documents of a website. The rules discovered are then used by WebTool to improve a website's navigation by suggesting links to popular documents as a user navigates through the website.

Jiang, Song and Zeng (2005) proposed an algorithm called the “co-visiting mining algorithm”. The algorithm assumes that documents accessed by users issuing the same queries represent a relationship. This information is then employed by a search engine to assist a user by guiding him to documents visited by other users’ with similar query. On the other hand, Agarwal’s (2004) collaborative crawling presents a way of mining web logs for frequent transitions in users’ visits, which can be used by a crawler to perform effective topic-based resource discovery on the Web. The method’s key advantage is a learning function that reuses information gained from one crawl to improve the effectiveness of later crawls automatically. Collaborative crawling extracts collective user behaviours from web logs to identify frequented documents belonging to a particular topic class in the Web. It assumes series of documents which are repeatedly visited by a large number of users accord to a coherent topic (interest) in users’ mind and clusters them together. This information allows a crawler to focus its crawling activities to documents that are topically related to a user’s current interest while avoiding links that are unrelated and no longer accessible.

Another interesting application of the web usage mining techniques is Cooley, Tan and Srivastava’s (2000) WebSIFT. WebSIFT is very similar to WEBMINER and the co-visiting mining algorithm, in that it also looks at mining and identifying frequent itemsets (documents that co-occur in users’ visits) from web logs. However it does not entirely rely on information gained from mining web logs as all the above applications do. Instead it demonstrated how content and structure information of a website could be used in combination with the web logs to provide evidence of existing relations between documents. Although web logs contain considerable amount of data with regards to website usage statistics and users behavioural patterns, the types of information that could be obtained from mining logs are still limited. For example, while being able to identify co-occurring documents from users’ visits and inferring that they might be related, it is difficult to provide physical or contextual evidence of this relation based on web logs. However the hyperlinks between documents provide structural evidence that they may be related, while website content information provides evidence on the contextual and topical similarity between documents. WebSIFT uses this additional information to automatically identify itemsets that are interesting. The discovered knowledge can be then used for design tasks such as restructuring a website's organization.

In this thesis, we will focus on mining web logs to extract navigational paths frequently taken by users browsing the school's website. Then we will look at employing data mining techniques to cluster groups of similar navigational paths together and find documents that frequently co-occur in each group. The frequently co-occurring documents are then used to suggest potential shortcuts, which could guide other users in navigating the website.

2.2 Other Forms of Web Mining

There are sources other than the web logs from which information regarding a website could be obtained, such as web content and structure. This section provides a brief overview on these two other forms of web data source and how they could be exploited using web mining techniques. The section also provides examples of web applications and highlights a few selected existing studies that were undertaken for both web content and structure mining.

2.2.1 Web Content Mining

Documents in a website contain textual information which is highly unstructured and varies accordingly to the purpose of a website. Unlike a highly structured database, retrieving information from a collection of documents is a challenging task as the documents do not share a common content representation. Furthermore there is no standard range of attributes available that can be used to distinguish one document from another in a website. In order to extract information from documents, a suitable document representation model is required. One such approach is the bag-of-words (BOW), commonly used to represent a document as a large collection of words. In BOW, a document's text is broken up into individual tokens or terms. Words that best distinguish a document is then determined by ranking terms using a frequency-based method (e.g., Term Frequency Inverse Document Frequency).

Web content mining involves applying data mining techniques on text contained in documents to extract useful knowledge from them, such as the topic relations between documents. This particular knowledge is commonly used for document classification. Document classification involves creating a topic hierarchy based on all documents of a website by categorising each document into a topic class it fits best. The topic hierarchy built can then be used for a variety of applications such as web browsing and web crawling. The majority of current studies rely on a manual classification technique which, requires feedback from users on each document's topic category and its importance to their information needs. However, manual classification of documents is a tedious and time-consuming task. It requires a reasonable number of feedbacks to generate a topic hierarchy and many users do not have the time to tag every document they visit in a website. This suggests the need for a method to automatically discover a document's topic and its relevance to a user's need.

Godoy and Amandi (2006) addressed the above issue with their Web Document Conceptual Clustering (WebDCC) technique. The technique utilises an incremental clustering approach that learns a topic hierarchy from the content of documents by combining the use of linear classifiers with k-Nearest Neighbour (kNN) algorithm. Initially, documents are preprocessed into BOW and are represented with weighted feature-vectors in space. WebDCC starts with a root node and creates new child nodes whenever it encounters a document that could not be classified under the existing nodes of the topic hierarchy. Each node in the hierarchy has a separate linear classifier and is associated to a topic category represented by a set of words that best describes it. A feature selection method is used to choose the set of words that describes a category at each node. At each node, the linear classifier constructs a prototype instance which is used to compare and decide whether or not a document belongs to the current topic category. The advantage of this technique is that it is an incremental topic learning mechanism that avoids the need for explicit user feedback in document classification. However, WebDCC is computationally expensive as it needs to explores all documents in a website to generate the topic hierarchy.

Crescenzi, Merialdo and Missier (2005) introduced an algorithm that performs document classification by identifying documents that share a common presentation template in a website. In this study, the authors argued that a large number of websites contain documents which have highly structured regions consisting of collection of links grouped together. They have observed that links found in such regions usually point to documents with similar layout and topics. Their algorithm captures this information as a schema consisting of a set of link collections found in the document. Documents with similar schema are grouped together and clustered into a candidate topic class. This candidate topic class is then added as a new node or merged into existing nodes of the topic hierarchy using the Minimum Description Length principle. Furthermore in contrast to the WebDCC, this algorithm only explores a small number of the website's documents to build the main categories of a topic hierarchy in order to reduce computational cost.

2.2.2 Web Structure Mining

A website consists of a collection of documents interconnected by hyperlinks. Each document may have a number of outgoing and incoming hyperlinks. A hyperlink from document A to document B may represent the continuity between units of information in a website. Figure 2.2 shows a graph representation of a website, where each node constitutes a document and the edges between the nodes are the hyperlinks connecting the documents of the website. An edge between any two nodes in the graph suggests that the documents may be related and may contain relevant information. Alternatively, the directory structure of a website may also represent relations between documents. Documents of a website are generally organised into hierarchical system. Documents containing similar information are often grouped together under the same folder. The assumption is that the documents that are grouped together indicate relations among them which, can be exploited with data mining techniques.

Web structure mining applies data mining techniques on the hyperlink structure and extracts information that can be used for a variety of purpose such as web crawling and ordering a web search result. Crawlers are automated programs that use the hyperlink structure to browse the Web in a systematic manner. A crawler starts with an initial list

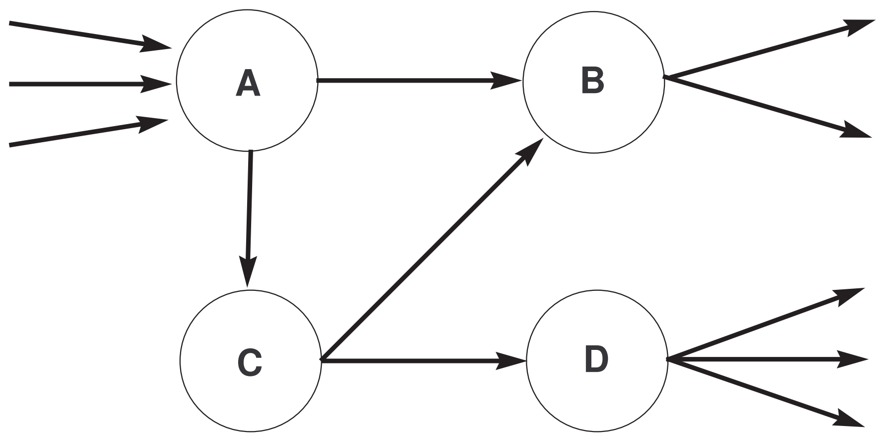


Figure 2.2: Website as A Graph

of URLs to visit and cycles through the list visiting each document in turn. While visiting a document it identifies all hyperlinks and adds them to the list of URLs to visit. This task is carried out recursively based on a set of selection policies for later processing by indexes.

A search engine uses the information from crawlers to find documents that match a query. The efficiency of a search engine can be measured by the proportion of relevant documents that are actually retrieved (recall) and the proportion of documents retrieved that are actually relevant (precision). Taking into consideration that users' information needs vary significantly from one individual to another, a search engine tries to retrieve all relevant documents. However, the Web spans over numerous websites and it is impossible to search the Web's entire document collection. Furthermore, a search engine can only retrieve a fraction of documents within a given time in response to a query. Therefore, it makes more sense to retrieve just a few selected important documents from a website, rather than its entire collection. This introduces the need for a way to identify and retrieve only the important documents of a website.

Page, Brin, Motwani and Winograd (1998) addressed the issue of distinguishing between important and non-important documents with their PageRank method that computes a rank for every document in the Web. PageRank is based on the citation count technique which uses the number of citations to a publication as an indicator of its importance. The more citations a publication receives the more important it becomes. Similarly, PageRank uses the incoming hyperlink structure as citations to a document from other documents to approximate the overall importance of a document. Figure 2.3 shows that *course.html* is a highly referenced document as it has a large number of incoming hyperlinks rather than *contact.html*. Furthermore, documents with hyperlinks to them from a highly ranked document have a higher importance compared to documents with hyperlinks from obscure documents. For example, in figure 2.3, *contact.html* is ranked higher than *research.html* although it has a lower hyperlink count than the latter because its referring page (course.html) has a higher importance.

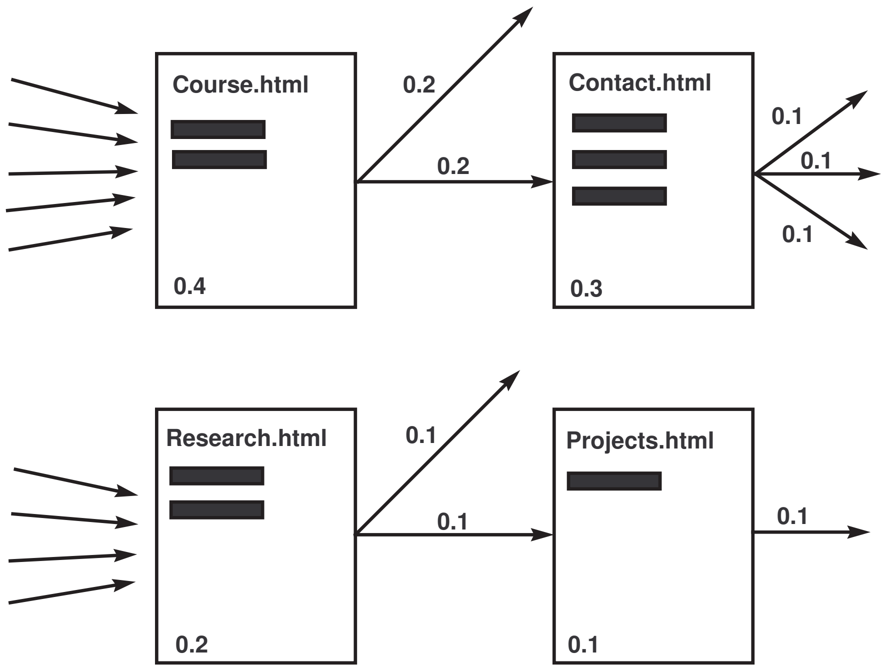
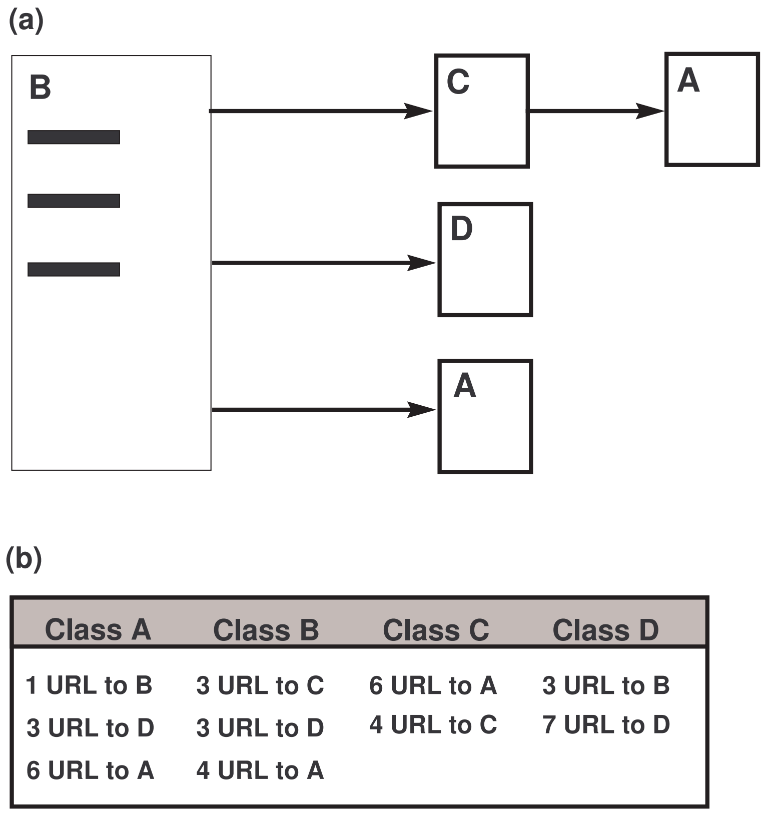


Figure 2.3: The PageRank Measure

The rank of a document is computed by calculating the sum of the ranks of its incoming hyperlinks as shown in figure 2.3. This information can then be used by a crawler or search engine to crawl or retrieve the documents of a website in the order of their computed importance. PageRank is highly efficient in measuring a document's importance. Nevertheless, a highly ranked document assumed to be of interest may not necessarily be topically related to a user's need. This is because the ranks assigned to documents are an indication of their importance in a website rather than to a user's need.

Later studies into web structure mining began employing hybrid methods which, included the use structure and content mining techniques to find interesting documents that are topically oriented to a user's need. Chakrabarti, Dom and van de Berg (1999) introduced a technique called Focussed Crawling to perform topic specific web crawls. Focussed crawlers tend to visit documents related to a specific topic rather than visiting the entire Web. The advantage of the technique is a pre-defined topic hierarchy that is used to train a naive-Bayesian classifier. Whenever the crawler encounters a document the classifier is invoked to categorise the document into a topic class where it fits best. The documents matching a target topic class are added onto the list of URLs to visit. This information is then used by search engines to guide future search in a similar topic by limiting the search space to a set of identified relevant documents.



Class A	Class B	Class C	Class D
1 URL to B	3 URL to C	6 URL to A	3 URL to B
3 URL to D	3 URL to D	4 URL to C	7 URL to D
6 URL to A	4 URL to A		

Figure 2.4: Rule-based Focussed Crawling

Alternatively, in (Altingovde and Ulusoy, 2004) the authors highlighted how focussed crawling overlooks the possibility that irrelevant documents may lead to topic relevant documents, which is referred to as tunnelling. They proposed a rule-based method that addresses this limitation. A naive-Bayesian classifier is trained using a pre-defined topic class hierarchy and a set of example documents for each topic class. The technique's advantage is a rule generation mechanism that associates a probability count to each topic class of the hierarchy. For each topic class, the documents referred to by the class's example document set are gathered. Rules are then formulated in the form of $T_i \rightarrow T_j(x)$, where x is the probability of a document of class T_i pointing to a document of class T_j. The probability is computed as the ratio of gathered documents in class T_j, to all gathered documents that the T_i class example document set refers to. These rules are then used to guide the crawling process.

Consider a website containing documents A, B, C and D as shown in figure 2.4a, in which each document constitute to a class with class A acting as the target topic class. Document B has links leading to documents C, D and A while, document C has a link leading to document A. In focussed crawling, the link leading to A is added to the URLs to visit while C and D being irrelevant is ignored. However, rule-based crawling consults each document's topic class probability rule (as shown in 2.4b) and finds that C does lead to the target class A with a probability of 0.6 and adds it to the front of the URLs list. This allows the rule-based crawler to perform deeper searches into a website's hyperlink structure for relevant documents, as opposed to conventional crawler's breadth first search method. However, this feature increases the computational cost involved in crawling for documents. Both focussed and rule-based crawling rely heavily on users' input to generate a topic hierarchy that can be used to pre-process and categorise the documents. Nevertheless, these two techniques demonstrate the advantage of combining the use of structure mining and content mining to improve the performance of web applications.

2.3 The Evolution of Websites

This section aims to provide a general idea of the developments in Web technologies that have taken place in the recent years. Firstly, we will be highlighting on the factor that motivated the developments and the resulting changes to the websites trend. This includes the introduction of different types of intelligent websites technology and identification of several successful existing implementations related to our study. The section goes on to define the Adaptive Websites terminology and discusses how it differs from existing intelligent websites. It also provides a review on a few selected studies to highlight the developments of adaptive websites. A detailed discussion of the advantages and disadvantages of the identified related studies in this section forms the motivation for our own work which will be elaborated further in the following chapter.

2.3.1 Static Websites

In the early years, the Internet mainly consisted of websites that were static in nature. A static website is a website that upon creation does not change. All aspects of the website such as its content, presentation and structure is determined by the web developer prior to publishing and it remains the same, until the developer decides to republish a new version of the site. Static websites have a very basic design and simply deliver information to users. This makes it very easy to develop and cheap to host and it requires almost no maintenance at all. Even now small companies are still relying on static websites to make their web presence. However the downside to a static website is that it is difficult to update and usually needs the expertise of a web developer to perform updates. This increases the overhead cost involved in maintaining the website. Furthermore the content and design of a static website can get stagnant over time. More importantly, such websites are catered towards a company's requirements rather than focusing on what the users' want. In any business, success and growth is usually defined in proportion to the demands generated by customers for the services provided. This suggests that it is vital to know what a customer needs in order to provide satisfactory services which can create more demand and increase the business opportunities. This brought about a need for more intelligent websites which could relate the information available in a website to the different preferences of users in the Web.

2.3.2 Intelligent Websites: Modelling The User

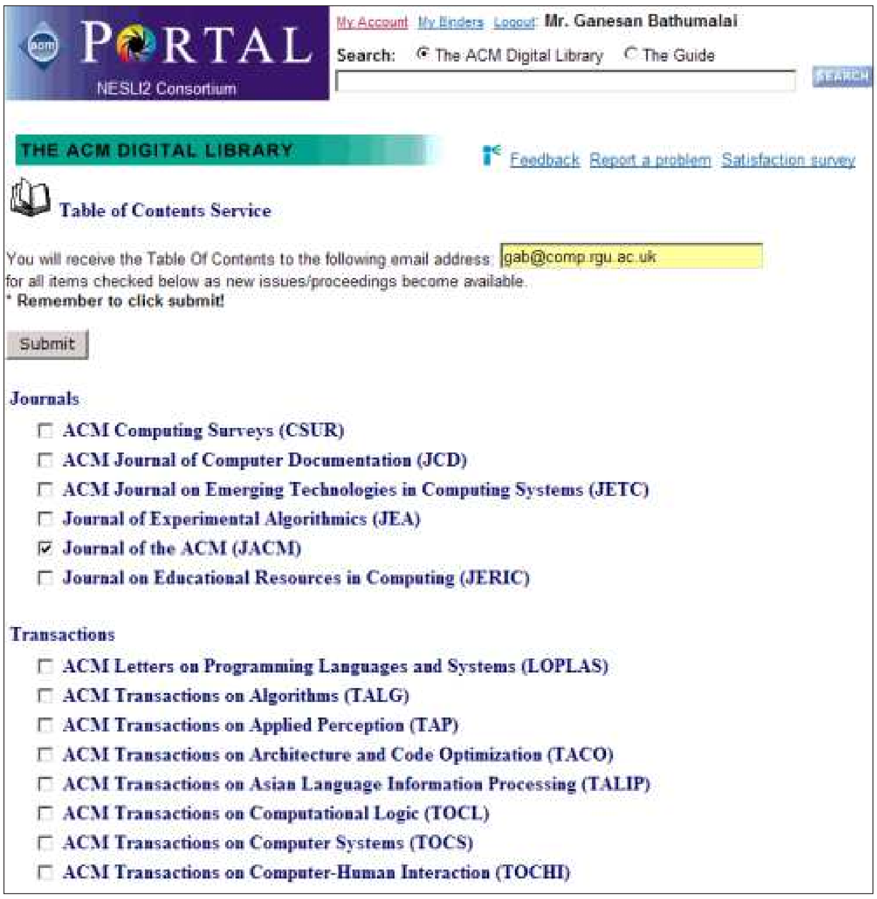
The rapid and constantly changing browsing needs of today's users in the Web generated demands for more interactive and user-centred websites. This acted as the catalyst for the developments of intelligent websites which could cater for the different users' needs. The foremost applications of intelligent websites date back to the mid 1990's and were focussed on modelling individual user behaviours. Different forms of intelligent websites have emerged since 1990 and the following subsections will discuss some of these forms in detail, from the viewpoint of user modelling.

Customisation

Customisation is the earliest form of intelligent website application available on the Web. It involves changing a website's interface and content to match a user's preference. Customisation is inspired by the idea of user modelling to overcome information overloading in a website. User modelling or user profiling involves gathering information about a user's preference (likes and dislikes) on the Web. Different users will have different view of the data available in a website. Therefore users are likely to have diverse preferences on the type of information that they would like to view during their interaction with the website (Langley, 1999).

Customisation allows a user to control the look and feel of the interface and the type of content provided by indicating ones preference through personal profiles. A user's profile is commonly gathered through measures such as form filling either at the beginning or at the end of the user browsing session. Each user identifies his interests from a set of predefined interest category or provides an objective description of his interests. This knowledge is used to configure a website's content by filtering out information irrelevant to a user's interest.

The *portal.acm.org* website is a typical application offering a customisation service to users. It contains a huge collection of electronic journal publications in various computing topics. In *portal.acm.org*, a user specifies the preferred journal topics (as shown in figure 2.5). The website then identifies journal articles matching the user's choice of topic and


PORTAL
NESLI2 Consortium
My Account My Binders Logout
SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey
Submit

Mr. Ganesan Bathumalai

Search: [✓ The ACM Digital Library] The Guide

Table of Contents Service

You will receive the Table Of Contents to the following email address: gab@comp.rgu.ac.uk
for all items checked below as new issues/proceedings become available.
* Remember to click submit!

Journals

- ☐ ACM Computing Surveys (CSUR)
- ☐ ACM Journal of Computer Documentation (JCD)
- ☐ ACM Journal on Emerging Technologies in Computing Systems (JETC)
- ☐ Journal of Experimental Algorithmics (JEA)
- [✓ Journal of the ACM (JACM)]
- ☐ Journal on Educational Resources in Computing (JERIC)

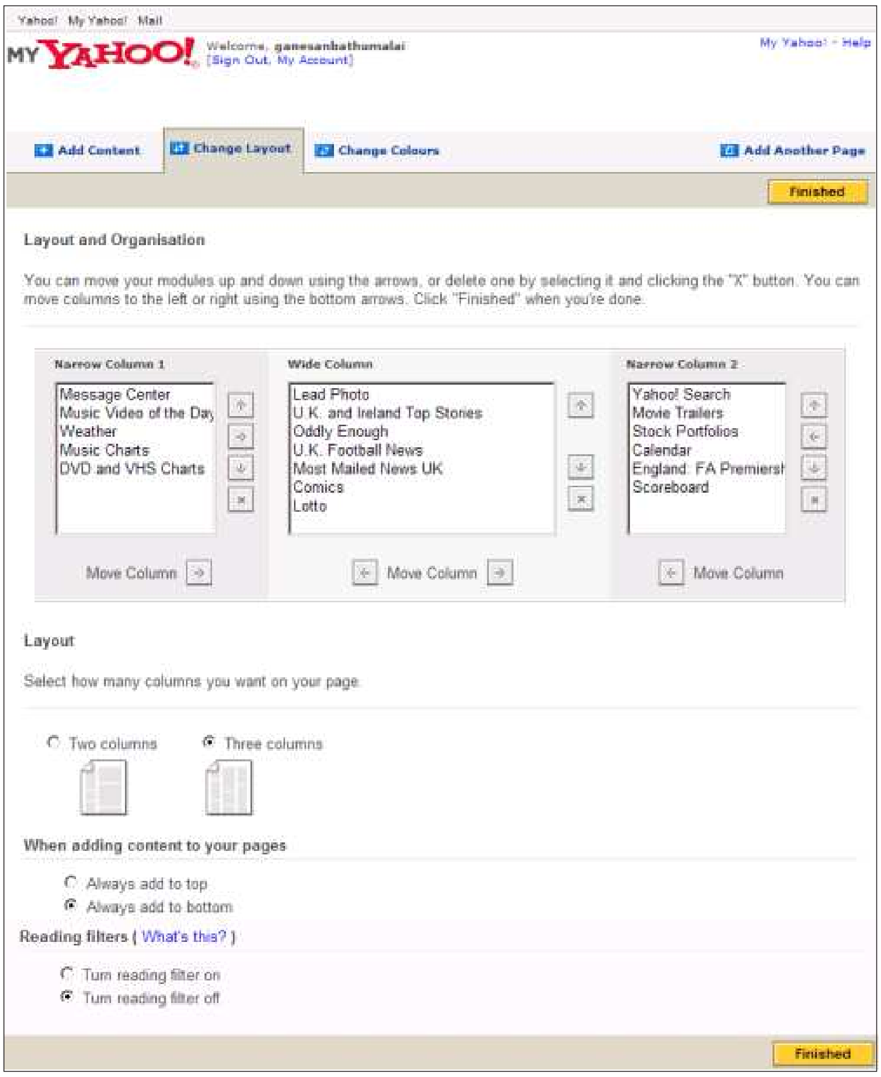
Transactions

- ☐ ACM Letters on Programming Languages and Systems (LOPLAS)
- ☐ ACM Transactions on Algorithms (TALG)
- ☐ ACM Transactions on Applied Perception (TAP)
- ☐ ACM Transactions on Architecture and Code Optimization (TACO)
- ☐ ACM Transactions on Asian Language Information Processing (TALIP)
- ☐ ACM Transactions on Computational Logic (TOCL)
- ☐ ACM Transactions on Computer Systems (TOCS)
- ☐ ACM Transactions on Computer-Human Interaction (TOCHI)

Figure 2.5: ACM Customization Service

alerts him only to the set of matched articles. Similarly, MyYahoo website allows a user to select content and customise its layout on the screen from a choice of modules as depicted in figure 2.6. The modules include stock prices, news items, weather reports and others for which the content is updated regularly. In addition, the users can also determine the frequency of content update.

The customisation approach makes information available upfront and improves a user's browsing experience by reducing the time spent on browsing irrelevant documents. Nevertheless, conventional customisation requires the users to exert effort and time in setting up a complex customisation feature by filling in forms. In addition, a user's profile remains static and does not change with the user's changing needs unless the user puts in effort to update it.



Layout and Organisation

You can move your modules up and down using the arrows, or delete one by selecting it and clicking the "X" button. You can move columns to the left or right using the bottom arrows. Click "Finished" when you're done.

Narrow Column 1	Wide Column	Narrow Column 2
Message Center	Lead Photo	Yahoo! Search
Music Video of the Day	U.K. and Ireland Top Stories	Movie Trailers
Weather	Oddly Enough	Stock Portfolios
Music Charts	U.K. Football News	Calendar
DVD and VHS Charts	Most Mailed News UK	England: FA Premiersh
	Comics	Scoreboard
	Lotto	

Layout

Select how many columns you want on your page

Two columns [✓ Three columns]

When adding content to your pages

Always add to top
[✓ Always add to bottom]

Reading filters (What's this?)

Turn reading filter on
[✓ Turn reading filter off]

Figure 2.6: MyYahoo Customization Service

Personalisation

Personalisation, is an automated website customisation technique that employs an implicit user profiling method to change the appearance and content of documents to suit the user. In personalisation a website monitors a user's browsing behaviour and reacts by providing content based on his interaction with the website. The user is not directly involved in content selection or interface configuration unlike conventional customisation. Instead, personalisation uses web usage mining techniques to automatically procure a user's profile from web logs. Automated user profiling involves gathering information about a user's activity on a website such as tracking links visited and discovering his navigational route through the website and time spent within documents, without interrupting his browsing session. This allows a website to keep track of a user's changing needs while putting less extra work on him, as opposed to the tedious form filling method.

Lieberman's (1995) Letizia, is a website personalisation tool that provides assistance to a user browsing a website in search of interesting information. It employs a personal intelligent agent to automate the task of user profiling for website personalisation. The agent learns by observing a user's interaction with the website and uses his past browsing behaviours to create a model of the user incrementally. This model is then used to infer documents that might be of interest to the user. Another interesting aspect of Letizia is the approach it uses to determine a user's interest. Letizia uses the link traversed by a user as an indicator of his interest. The author argues that following a link visited by a user is fairly a good indicator of his interest in the document containing the link. In addition, repeated return to the same documents by a user further reinforces the above assumption. Lieberman's (1995) agent-based approach to user profiling eliminates the need for explicit user feedback to perform effective website personalisation.

Godoy and Amandi's (2006) PersonalSearcher personalises web search result by filtering documents corresponding to a user's interests from a set of retrieved documents during a web query and presents the information in a separate window. It takes a slightly different approach to automate the task of user profiling. PersonalSearcher learns meaningful user profile by extracting concepts from documents visited by a user in a browsing session. A personal agent then uses the profile to classify documents into different categories of a user's interests and returns the documents matching the same category of the current user profile. Most importantly PersonalSearcher uses the time spend on reading and browsing a document with respect to its length as an indicator of a user's interest. The assumption is that a user usually spends longer time browsing an interesting document compared to an irrelevant document, which is argued to be equally a good indicator of a user's interest in a website.

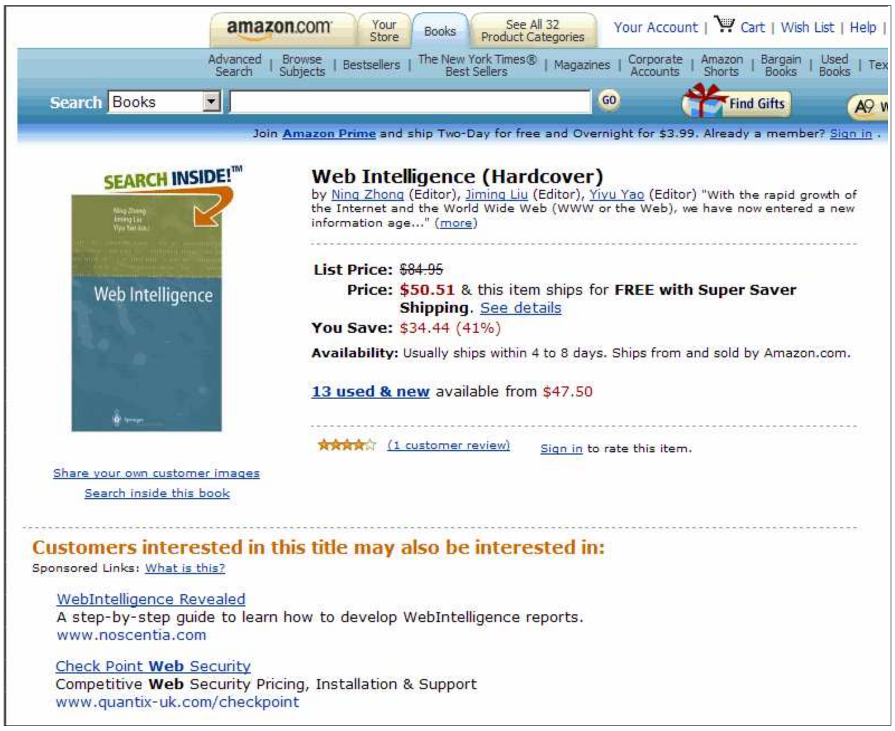
Personalisation services need not necessarily be on an individual basis as exhibited in the above studies. The interface and content of a website can also be tailored to a predefined group of users based on their previous browsing experience. Group personalisation involves identifying user groups and the type of content that is appropriate for those groups. It employs web logs mining technique to perform group profiling which, involves characterising and segmenting the users of a website based on a specific grouping criteria. There are different kinds of grouping criteria's available for example by user's interest and by organisation. Lieberman, Dyke and Vivacqua's (1999) "Let's Browse" uses an agent to guide a group of users to browse a website by suggesting information of common interest to the group members. It is based on the idea that despite having distinct needs, some users may share a common topic of interest and it is useful to group them together to perform website personalisation. Users' profiles are pre-gathered off-line from web logs and transmitted to the system through an electronic badge called "meme tag", which is worn by each user. The system detects the presence of users' through the tag and constructs an interest profile for a group of users from the set of users' profile currently present. It then browses the website and displays a selection documents that the agent thinks might be of mutual interest to the members of the group.

Users can also be grouped based on the organization they are attached to and different sections of the website are presented to each user group identified. This is known as organisational personalisation and is very much similar to group personalisation but focuses on the needs of a large community of users from an organisation, rather than individual user groups. It has been observed that users from the same organisation tend to visit similar documents specific to their organisation's aim and interest. Rangarajan, Phoha, Balagani, Selmic and Iyengar's (2004) clustering algorithm uses the above observation to perform website personalisation. The algorithm uses an unsupervised clustering technique called adaptive resonance theory that groups users into distinct clusters by learning from their web access patterns. Initially, the algorithm distinguishes and retains host names that have made large document requests assuming such host represents a large community of organisationally related users. The algorithm then represents each user request recorded in the web logs as a feature-vector which is used to identify the cluster (i.e., organisation) a user belongs to. A prototype vector is then computed to represent each cluster by generalising the URLs most frequently visited by all members of the cluster. Therefore each cluster is associated to a set of documents that represent the access patterns and interests of a large community user group. This knowledge is then used to personalise a website's information provided to the users from the same organisation.

Recommender Systems

The automation of user profiling significantly increased the development of intelligent website systems in the Web. These systems rely on individual user profiles in order to group users and perform website personalisation. However, it is virtually impossible and impractical for developers to maintain profiles for the rapidly increasing number of visitors to a website. Following this many research shifted to a non-profile based approach to provide website personalisation, giving rise to a new range of intelligent website systems known as the recommender systems. A recommender system applies web mining techniques on web logs to gather users' interests and comes up with a set of personalised recommendations to interesting information in the website. For example, Mitchell, Joachims and Freitag's (1997) WebWatcher uses web usage mining technique to discover documents that is of interest to users and recommends links to them. Similarly Mobasher, Cooley and Srivastava (1999) applied usage-based clustering to identify frequently accessed URLs and performs real-time website personalisation by recommending links to documents related to users' needs as they browse a website. Apart from the above, other successful applications of recommender system include collaborative filtering and pre-fetching.

Collaborative filtering is a popular technique that is often used to develop recommender systems. The technique makes recommendation to users based on the preferences of the community of like-minded people. It is inspired by Shardanand and Maes's (1995) social information filtering approach that makes use of the word of mouth of users to suggest information of interest to other users on the Web. The approach exploits the similarities between interest profile of a user and those of other users in the database to make recommendations. Instead, collaborative filtering keeps no record of a user's profile and exploits users access patterns to make personalised recommendation to users seeking similar item of interest. The content to be delivered is chosen based on the assumption that a particular user will value an item that of which other like-minded people also enjoyed. A typical application of collaborative filtering is the Amazon website, which looks at the current product being viewed by a user and suggests links to products purchased by other users that has viewed the same product. Figure 2.7 shows Amazon's recommendations for the topic of "Web Intelligence". The technique depends on users link traversal information and disregards the content of the browsed items. As a result, not all of the system's suggestion may be related to a user's need.


amazon.com
Your Store
Books
See All 32 Product Categories
Your Account | Cart | Wish List | Help |
Advanced Search | Browse Subjects | Bestsellers | The New York Times® Best Sellers | Magazines | Corporate Accounts | Amazon Shorts | Bargain Books | Used Books | Tex
Search Books
GO
Find Gifts
A9 W
SEARCH INSIDE!™
Web Intelligence

Join Amazon Prime and ship Two-Day for free and Overnight for $3.99. Already a member? Sign in .

Web Intelligence (Hardcover)
by Ning Zhong (Editor), Jiming Liu (Editor), Yiyu Yao (Editor) "With the rapid growth of the Internet and the World Wide Web (WWW or the Web), we have now entered a new information age..." (more)

List Price: ~~$84.95~~
Price: $50.51 & this item ships for **FREE with Super Saver Shipping**. See details
You Save: $34.44 (41%)

Availability: Usually ships within 4 to 8 days. Ships from and sold by Amazon.com.

13 used & new available from $47.50

★★★★☆ (1 customer review) Sign in to rate this item.

Share your own customer images
Search inside this book

Customers interested in this title may also be interested in:
Sponsored Links: What is this?

WebIntelligence Revealed
A step-by-step guide to learn how to develop WebIntelligence reports.
www.noscentia.com

Check Point **Web** Security
Competitive **Web** Security Pricing, Installation & Support
www.quantix-uk.com/checkpoint

Figure 2.7: Amazon's Recommender System

Good, Schafer, Konstan, Borchers, Sarwar, Herlocker and Riedl (1999) addressed the above limitation using an agent solution with collaborative filtering to increase the quality of recommendations. A personal agent is used to maintain the preferences of a user as the user browses a website. The preference is subsequently used to filter the recommendations made to the user by the system. This allows collaborative filtering to determine the usefulness of information to each user before recommending it. Similarly Pavlov, Manvoglu, Giles and Pennock (2004) use a maximum entropy model to improve the recommendations of collaborative filtering systems. The insight to the model is a clustering algorithm used to mine for groups of topically related documents from past user browsing patterns. Clusters found are then used to inform the collaborative filtering mechanism to predict the next related document for recommendation based on the current document requested by the user. The advantage of collaborative filtering is its speed and efficiency in computation, which provides rapid feedback. However, it relies on a critical mass of users to make useful suggestions and a small user population leads to lower quality recommendation. In addition, it is unable to make a recommendation to a user if a match with like-minded users' cannot be found.

Pre-fetching is another form of recommender system that is gaining popularity in the Web. It uses the knowledge acquired from web logs to make recommendation by guessing the next page the user is likely to visit based on the current page being viewed. This allows a client to pre-load the document into a cache and serves the document much faster upon a user's request. The development of pre-fetching applications plays a vital role in the Web to reduce user inactivity during a document request between the client and the server (Fan, Cao, Lin and Jacobson, 1999). In addition, it has been widely applied to reduce the load on network bandwidth placed by enormous number of requests to the server each day by millions of users. Rangarajan et al. (2004) introduced a system that uses usage-based cluster mining technique to mine set of URLs that are commonly accessed by a large community of organisationally related users (refer to section 2.3.2). Each user group is represented by a cluster of URLs frequently accessed by them, mined from the web logs. Whenever a user browses the website, the system automatically identifies the group that user belongs to and retrieves the set of URLs associated to the group. Based on the current document being viewed by the user, the system then predicts documents that the user may request. These documents are then preloaded into the server cache to be served instantaneously to the user upon his request.

2.4 Moving Towards A New Challenge: Adaptive Websites

Earlier intelligent websites are inclined towards information filtering, path prediction and recommendation. While these forms of personalised services reduce the difficulties involved in browsing and navigating through a website, it does not address the root of the problem, which is design related. Furthermore, website personalisation has recently been criticised as an expensive, ineffective and unproductive practice of the Web. According to Festa (2003) and McGovern (2003), it costs more to operate a website offering personalised service than a normal website. The cost includes the additional human efforts required to manage and maintain the system and the extra licensing cost for the software. These factors lead to the introduction of adaptive websites. The term adaptive websites was introduced by Etzioni and Perkowitz (1997) and defined as websites that are capable of changing their organisation and presentation based on users' access patterns. Developing an adaptive website involves transforming a website's design based on collective users access patterns and website usage statistics. Thus indirectly it can be regarded as an act of modelling the website in contrast to modelling the users.

The motivation for our work is derived from the idea of the transformation approach (i.e., modelling the website), where we are aiming to suggest changes to the school's website design based on the collective users access patterns obtained from the school's web logs. In other word, we are looking to suggest changes to the actual organisation of a website rather than performing superficial changes to the site as personalisation does. The rest of the section is organised as follows. Section 2.4.1 defines the transformation approach and provides a detailed explanation of its aspects. It also identifies the different types of transformational changes there are and its application to a website. Section 2.4.2 and 2.4.3 are dedicated to discuss the two forms of application of the transformation technique. In particular, the sections explore related studies that use transformation techniques to develop adaptive websites and identify their limitations. It is these limitations that give form to our own research problem and application area.

2.4.1 Transformation

Transformation is an approach that focuses on the basics of a website design such as the ease of use and navigation. These days many websites have incomprehensible content and confusing navigation. A flexible and simple to use website navigation gives the users a sense of satisfaction and adds more value to their browsing experience. When a user visits a website, the user seeks the simplest and easiest way to enter a website, access the required information and leave without exerting a great deal of effort. The transformation approach aims to provide just that, by altering a website's organization or presentation or both based on users' access patterns and website usage information. In addition, whereas personalisation focuses on individual or group customisation, transformation seeks to change a website for all users.

The changes made to a website can either be destructive or non-destructive. A destructive change is an act of altering a website's design which involves the removal of information from the website. For example, removing documents or sections of a document, deleting hyperlinks and reorganising the website's structure are regarded as destructive website transformations. Non-destructive changes include any act of altering a website's design without the removal of information from the website. For example it may involve tasks such as highlighting hyperlinks within documents, adding new hyperlink between documents and reorganising information in a document. The changes that can be made to a website are highly influenced by an organisation's policy towards the manipulation of information in their website.

There are three ways in which transformation can be performed: manual, semi-automated and automated. In manual transformation the human is in control. The system generates a set of proposed website changes and conveys it to the website's administrator. The administrator then reviews the proposed changes and decides whether to implement them or not. A semi-automated transformation allows the system to propose and implement non-destructive changes to the website. However, any destructive changes that involve in the deletion of information from the website require the approval of a human administrator. Finally, in automated transformation the system is given full control to propose and implement both destructive and non-destructive changes to the website as appropriate without human intervention.

There are a number of issues that need to be taken into consideration when transforming a website. The first issue involves determining what aspect of the website is to be transformed and how is it going to be represented. The second issue is about identifying where and how to implement the proposed adaptations to the website. And the final issue is deciding how frequent the adaptation should be made to the website. There are two ways in which the transformation technique could be applied to a website. The first focuses on transforming the website's organisation (i.e., organisation adaptation) while, the second involves transforming the website's presentation (i.e., presentation adaptation). The following subsections describe each of these in detail and identify a few prominent studies which, propose various ways to transform a website's organization or presentation.

2.4.2 Organisation Adaptation

Adapting a website's organization involves altering the way information is organized within the website. The underlying idea is to identify items that are frequently accessed by users and making them easier to be found. This allows users to save time and effort in trying to find a piece of information that may be buried deep within a website's structure. As mentioned earlier, there are a number of critical design related issues that need to be addressed prior to making any kind of adaptation to a website and they vary according to the kind of transformation applied to the website. For example, if the transformation involves adding hyperlinks to a website, the first issue would be how to represent the hyperlinks meaningfully in the website, secondly where and how to add them, thirdly how many hyperlinks should be added and finally how frequently they should be added.

Perkowitz and Etzioni (1997) introduce a non-destructive organization adaptation called "promotion and demotion" that makes a hyperlink or a document easier to be found by placing a reference to it in the index document. A promotion score is used to determine which items (hyperlinks or documents) need to be promoted and demoted. The score is computed as the popularity of an item over its accessibility. The popularity of a hyperlink is determined by the number of times it was traversed and the popularity of a document is given by the number of times it was accessed. Meanwhile, the accessibility of a hyperlink and document refers to how far down the document a link is and how far away a document is from the index document. Items with high popularity but low accessibility are promoted in a dedicated dialogue box with limited space that appears in the index document. Whenever a highly popular item is added (promoted) into the dialogue box, items with low popularity will be removed (demoted). This makes important information easier to find while unpopular information are moved out of the way. Nevertheless, this study assumes visitors enter a website via the index document and focuses all promotion there. Users entering the website via documents other than the index document will not benefit from the promotions.

A more sophisticated form of organization adaptation is linking, which is also referred to as shortcutting. Shortcutting involves adding a new link between two documents which was unlinked in the website based on correlations in users' visits. The assumption is: If two unlinked documents frequently appear together in many users' visits, it suggests that they are topically related in the users mind. Therefore it is worthwhile adding a link between them. Shortcutting is commonly used to reduce the number of links a user needs to click, in order to find a target document (a target document being a document containing information of interest sought by a user).

Srikant and Yang (2001) propose a non-destructive website adaptation algorithm that adds useful shortcuts to target documents in a website. The authors claim that users' expectation of where a target document might be located in the website differs significantly from its actual location. Thus providing a link from the expected location to the target document reduces the time spent by users browsing for it. Target documents are identified using a time threshold set by the webmaster. The documents a user spends more time in than the threshold are considered to be target documents, while the expected locations of a target document are identified using the point of backtrack in users' traversal paths extracted from web logs. For example, figure 2.8 shows the hierarchical structure of a website and a user's traversal path in search of the target document F.

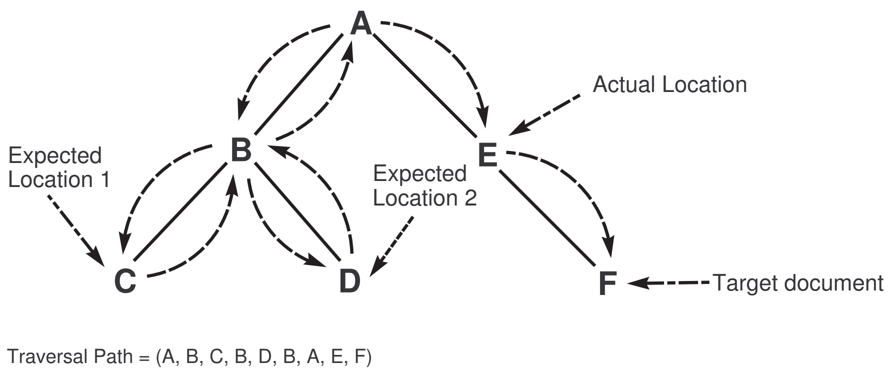


Figure 2.8: Expected Location and Actual Locations of Target Documents

Here the user starts at the root document A, goes to B expecting to find document F under it, and then to C. Realizing F is not under C, the user backtracks to B and tries under document D before backtracking all the way to the root and finally finds F under E. Thus the expected locations for target document F is C and D, while the actual location is E. The algorithm then sorts all expected locations for a target document based on their visit frequencies and suggests shortcut to be added from the most frequently visited expected location to the target document. The drawback of this algorithm is that there is a tendency that the algorithm may mistake target documents for expected locations and vice versa. This could result in users' being misguided to the wrong documents in the website.

Similarly Lee and Shiu (2004) propose a destructive organization adaptation that improves the structure of a website based on users' access patterns by adding and removing hyperlinks in the website. In this study the authors claim that the longer a user's traversal path is, the higher the chances are for him to make mistakes while browsing for target documents. A traversal path is defined as a set of documents visited by a user to reach a target document and target documents are identified using a pre-determined visit duration threshold. According to the authors, a website's efficiency can be improved by suggesting shortcuts that reduce the length of the path a user needs to traverse to find the target document. A website's overall efficiency is estimated by averaging the sum of all user's traversal paths efficiency over the total number of users, whereas a path's efficiency is computed by dividing the shortest path over a user's operating cost. The shortest path is the actual number of documents a user needs to traverse to find a target document obtained from the website's structure, while user operating cost is the number of documents traversed by them to reach the target document.

Hyperlinks to be added and removed are determined based on the expected efficiency value of a path. The expected efficiency value is obtained by suggesting a shortcut between two documents in the path and recomputing the path's efficiency. If the expected efficiency (value after adding shortcut) is higher than the path's actual efficiency (value before adding shortcut) and the shortcut refers to a non-existing link, it is suggested to be added as it improves the website's efficiency. However if the expected efficiency is higher than the path's actual efficiency and the shortcut refers to an existing link, then the link is suggested for removal as it does not improve the website's efficiency. Lee and Shiu's (2004) adaptation method allows users' to access target document in less number of clicks and at the same time improves overall website navigation by identifying and removing ineffective links. A weakness of this study is that the authors do not give a clear account of how two documents in a path are selected for shortcutting.

Brickell et al. (2007) propose a non-destructive website adaptation algorithm that performs shortcutting between documents of a website and promotes them in the index page of the site. According to the authors, if a path to a target document is unusually long and winding, then users may never reach their intended target document or they will have to exert effort to do so. Thus by mining user access logs and using a caching algorithm, they are able to predict which document a user is likely to load next based on the current document being viewed by the user. Based on the above information, the system then provides a direct link between the two documents if the documents are unlinked in the website. Unlike existing shortcutting algorithms that propose shortcut based on offline data, their algorithm is able to suggest shortcuts in real time. However, the drawback of their shortcutting algorithm is that it skips over the intermediate documents in a path assuming that they are irrelevant just as other existing shortcutting algorithm does.

2.4.3 Presentation Adaptation

Presentation adaptation involves altering the way information is presented in the website. All too often users browsing behaviours change rapidly while the website remains static. The idea here is to monitor how information in a website is being viewed by the users and make changes to the way this information is presented to them based on the observation. This provides users with an enhanced browsing experience as the website continually adapts its presentation to reflect their changing browsing behaviour.

To our knowledge Etzioni and Perkowitz's (2000) IndexFinder is the only form of study to date that has attempted to adapt a website's presentation non-destructively based on users browsing patterns. The study investigates the problem of automatically synthesizing new index documents containing a collection of links to documents that are topically related but currently unlinked in the website. IndexFinder is a conceptual cluster mining algorithm that relies on visit coherence assumption to identify clusters of documents that share a common topic and presents them as content (i.e., collection of hyperlinks) for candidate index documents. According to the visit coherence assumption, a set of documents which is repeatedly visited by a large number of users represent a coherent topic in their minds and the documents are likely to be related. The key idea here is to find groups of documents that often co-occur together in users' visits, from which a candidate set of index documents can be generated.

IndexFinder uses a combination of a statistical cluster mining technique and a concept learning algorithm to identify clusters of topically related documents from web logs. A co-occurrence frequency measure is used to determine the similarities between pairwise documents and clusters frequently co-occurring documents together. Each cluster found comprising of set of related documents is then fed into a concept learning algorithm which produces simple concepts (i.e., conjunctions of attribute/value pair rules) that best describe what the documents share in common. Documents that do not match the given concept description are discarded from the cluster, while other documents in the website that match the concept are added to it. Finally, these clusters are ranked and the ten most highly ranked clusters are presented to the webmaster as a set of candidate index documents. The webmaster then decides whether or not to add the newly generated index documents to the website.

The advantage of the IndexFinder algorithm is that it generates index documents that reflect the way users view the information available in the website. In addition, it also improves users' navigation in the website. However, the algorithm's performance must be judged against a limited range of training and test data. The data used for training and testing are drawn from a small group of websites comprising of structured documents with distinguishable attributes, whereas the majority of websites on the Web consist of documents with highly unstructured content. Nevertheless, IndexFinder demonstrated the transformation that can be performed on a website's presentation by learning from user access patterns.

2.5 Chapter Summary

In this chapter, we have surveyed the broad area of Web Mining with a specific focus on web usage mining. The fundamental aspects of web usage mining such as the source of data and the challenges involved in extracting knowledge from web logs were explored in detail. Existing usage mining techniques and their potential applications to the Web were also discussed. Here, we have identified the advantages that could be gained from mining web logs. We also highlighted the limited types of information that could be obtained from web logs is a significant drawback. However, this can be overcome by using additional information from website content and structure. We have also briefly reviewed the two other forms of web mining, which are content and structure mining, highlighting their data sources as well as applications to the web. Although the review done on content and structure mining is not extensive, it is still possible to concur that the substantial work involved in them could amount to a mini project on their own. Therefore we have decided that our work will focus entirely on usage mining to obtain knowledge about the users and the website from web logs.

Following that, this chapter looked at the evolutions in Web technology from static to the current intelligent websites. We have seen techniques that use automated user profiling to perform website personalisation and those that rely solely on web logs rather than users' profiles to personalise a website. However, we realised that majority of the current intelligent websites were too focussed on modelling the users to satisfy their Web needs and fail to address the actual source of the problem which, is design related. Adaptive website can be seen as an optimization problem, where a website continually tries to maximise the efficiency with which users can browse and navigate through it. The difference between current intelligent websites and adaptive websites is that the former filters website information to be presented to a user, while the latter actually transforms it (i.e., alters the website based on users' browsing behaviour so as to reflect their changing needs), thus improving access to information. Nevertheless, the development of adaptive websites capable of transforming their design is still in its infancy. This is mainly due to concerns such as privacy and data loss.

Concern about privacy is one of the major factors affecting the development of adaptive websites. Many users have voiced their unease over the collection, use and distribution of their personal information (e.g., browsing behaviour) online as it intrudes into their privacy. Furthemore, some websites are known to employ unethical methods to gather personal information about users without their knowledge. This has caused users to be suspicious of the role of an adaptive website and abstain from such websites. Similarly, organisations rendering services to users online are skeptical about the true potential of a transformation based adaptive website. They fear that these websites may cause more harm and chaos rather than benefiting them by making detrimental changes to their website's design resulting in irrecoverable data loss.

Despite these uncertainties, a number of studies have demonstrated the feasibility of transformation-based adaptive websites. These studies as outlined above include promotion and demotion (Perkowitz and Etzioni, 1997), linking or otherwise known as shortcutting (Srikant and Yang, 2001; Lee and Shiu, 2004; Brickell et al., 2007) and index page synthesis (Etzioni and Perkowitz, 2000) which, demonstrated the potential benefit that could be gained from such intelligent website systems. The main advantage of the transformation approach is that it focusses on altering a website's design for the benefit of everyone rather than an individual.

The motivation for our work is derived from the transformation approach i.e., modelling the website for all users. Our work aspires to suggest changes to the school's website organisation based on the collective users access patterns obtained from the school's web logs. In order to achieve this, we chose to address the research problem of finding useful shortcuts that could be used to improve the task of navigating through the school's website for all users. By useful shortcut, we mean a new link between two documents that could shorten the navigational paths taken by users to reach their document of interest.

Chapter 3

Our Approach To Adaptive Websites Using Wayposts

The objective of this chapter is to introduce our approach to an adaptive website and to provide details of the work carried out in acquiring data from the web logs. In particular, we will highlight a research gap found in the area of adaptive websites and suggest our approach to address it. Following that, the chapter provides a discussion on the acquisition of data from web logs which is divided into three tasks, namely Web Logs Filtration, Crawler Logs Elimination and Sessionisation. The chapter gives an account of the nature of each task and elaborates on the methods employed to carry them out. In order to establish our discussion, a detailed description of the web logs which will be used consistently through this and later chapters for evaluation is provided.

The chapter is organised as follows. Section 3.1 lists the research gap identified and introduces our approach to addressing it. Section 3.2 gives an introduction to the web logs that will be used in our experiments. Section 3.3 provides a brief description about the three tasks of data acquisition with sections 3.3.1, 3.3.2 and 3.3.3 giving a detailed account of the processes involved in each of them. Finally section 3.4 summarises the chapter.

3.1 The Approach

A complex website design leads to an increased number of links that the users need to click before finding the specific document they are looking for, thus making website navigation an extremely difficult task. As mentioned in the previous chapter, linking also commonly known as shortcutting is a form of organisation adaptation which is used to transform a website's design. Shortcutting aims to reduce the number of clicks required in a user's navigational path to reach a target document, assisting the user in navigating through a website and thereby also improving the website's organisation. Target documents are desired documents of interest containing information that satisfy the users current needs. A path can be thought as a user's journey from an initial document to a target document in a website. The sequence of documents visited along this journey indicate the navigational route taken by a user to reach his intended target document.

Longer paths tend to pose a higher risk for users to stray off-course from their intended route, often resulting in them missing their target document, since each document contains a large number of links from which a user needs to select one that will bring him closer to his target document. This is similar to travelling in a trunk road which has several intersections and each of these intersections leads to another set of intersections and so on to different locations. Choosing the right link at each intersection (i.e., documents users need to navigate) allows a user to reach his target document quickly. Alternatively, providing a shortcut to the target document greatly reduces the number of intersections in a path and ensures the users could reach their target document with fewer clicks.

Existing studies (Srikant and Yang, 2001; Lee and Shiu, 2004; Brickell et al., 2007) tend to minimize the number of clicks required to find a target document by providing a shortcut between the initial and target document in a path. This approach assumes the sequence of intermediate documents appearing in the path is insignificant to a user's information need and bypasses them. However, we believe that these documents may contain crucial information corresponding to a user's need that leads him to the target document. Therefore, our work aims to explore this possibility and presents a way to reduce the number of clicks required to find a target document by identifying *wayposts*, which can act as navigational shortcuts based on frequently travelled users' paths. Wayposts are intermediate documents in a path that could act as a significant guide for users to find the target document. It is similar to landmarks on roads, which inform drivers that they are travelling on the correct route to their intended destination. Likewise, wayposts act as signs in a website reassuring users that they are progressing toward their intended target document, giving them a sense of satisfaction throughout the browsing session.

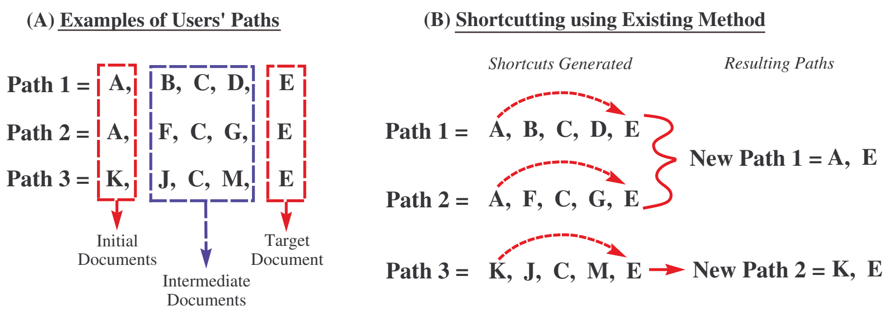


Figure 3.1: (a)Examples of Users' Paths (b)Shortcutting of Users' Paths

The advantage of our approach over existing methods is that it produces navigational shortcuts that reduce loss of significant information which may be contained in wayposts while allowing a user to reach a target document quickly. For example, consider the paths shown in figure 3.1a. All three paths lead to the same target document E with the first and second paths sharing the same initial document A. Each path requires 4 clicks in order to reach the target document. Existing methods would have provided a shortcut from the initial document to the target document bypassing the intermediate documents for all three paths as shown in figure 3.1b, thus reducing the lengths of the paths to a single click in order to reach the target document rather than 4 clicks.

Our approach seeks to identify important wayposts by grouping similar paths together. Grouping such similarly travelled paths together allow us determine documents that frequently co-occur in them. These documents which frequently appear in more than one path indicate users' interest in them, thereby making them a suitable candidate as wayposts. In figure 3.2 the first and second paths are grouped together as they are more similar, while the third path is grouped by itself. Shortcuts are suggested from the wayposts found in the groups as shown in figure 3.2. Our approach reduces the 4 clicks originally required to

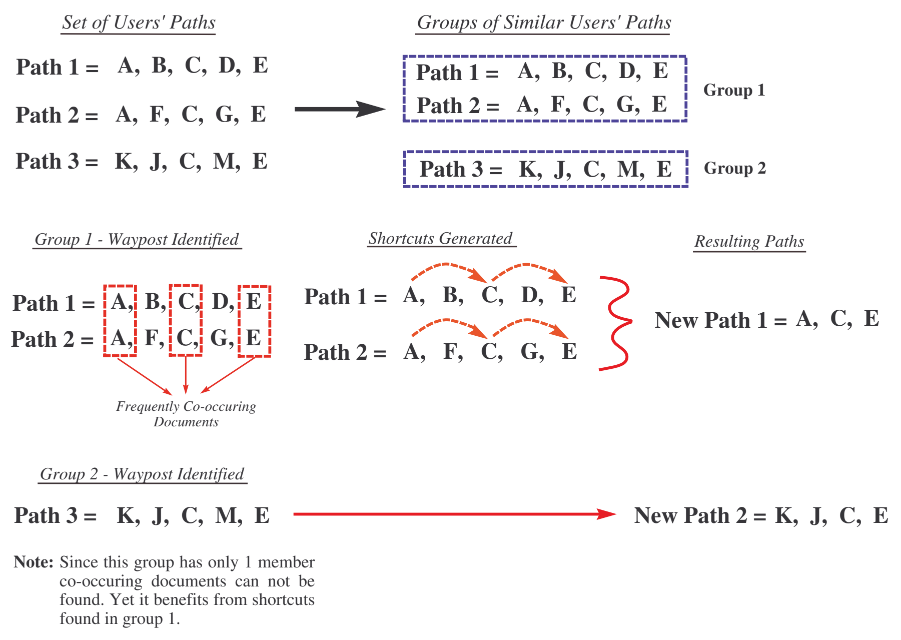


Figure 3.2: Our Approach

reach the target document E to 2 clicks for the first and second paths and 3 clicks for the third path. Note that although no wayposts could be produced from group 2, the path in it can still benefit from the shortcuts generated of group 1. As opposed to the existing methods, our approach enables users to take advantage of these new shortcuts which, still requires fewer number of clicks to reach their target documents while reducing the possible loss of information that may be contained within the intermediate documents of a path.

3.2 Logs Used

The web logs used for evaluations throughout this study is of the CLF type. These web logs are obtained from the in-house web server of the Robert Gordon University's School of Computing. The web server records users' activities on the School's website for every month over a period of six months interval. At every interval of the sixth month the old web logs are purged and replaced with newer ones. Our data is sampled from the month of October in the second half interval of the year 2004. The reason being that it is right after the start of the academic year 2003/2004 and probably has the most number of accesses made to the website by both external and internal users compared to any other month in that interval. External users refer to any client that are not directly associated with the University, while internal users are those attached to the institution such as the students and staff.

The October 2004 logs dataset consists of a total number of 1186935 document accesses (i.e., document requests) made to the School's web server by a total number of 5584 clients. Approximately 60-65% of the total number of accesses is made by internal users while the remaining number of accesses belong to external sources. For experimental evaluation in this study, we further created another two dataset from the October 2004 logs dataset. The first dataset consists of accesses made to the server from the 1^{st} to the 7^{th} of October 2004 and has a total number of 395058 document accesses made by 2200 clients to the web server. While the second dataset contains the user access logs from the 1^{st} to the 14^{th} of October 2004 with a total number of 862493 document accesses made by 4071 clients. These two dataset will primarily be used in the evaluation of the methods employed in later chapters. A sample of the October 2004 logs dataset can be found in appendix A for further reference.

Note that the figures above are not a generalisation of the total number of accesses a particular website could receive across the Web. It may well be a mere fraction of the actual number of accesses made to any website but it gives an approximate idea on the enormity of web logs and the vast knowledge that could be gained by mining them. The web logs used in this study presented several challenges in processing them into a form ready for usage mining. One intriguing challenge was to determine useful document accesses from unwanted accesses. The term useful here means document accesses that carry information which, could be used to infer or make informed judgement about users' browsing behaviours on the website. Removing unwanted data saves unnecessary computational cost however, due to the lack of information present in the web logs it is difficult to say what is important and what is not. Thus it could result in the loss of both important and unimportant data. Therefore, moderation is required during this process to avoid unnecessary data loss.

Another challenge presented by the dataset is the large number of document accesses that were made through proxy servers. These accesses are all associated to a single DNS/IP address being the proxy's therefore, making it difficult to isolate the document accesses made by different users. For example, all accesses made by the internal users of the School goes through the in-house proxy server and is assigned the same DNS/IP being "ROADKILL". This rendered nearly 65% of the dataset unsuitable for mining and all document accesses related to "ROADKILL" was excluded from the mining process. Therefore leading to more data loss and a possible bias as the remaining data are associated to external users only. However, the remaining 35% of the dataset yielded some interesting results, as it revealed to contain both internal and external users' document accesses. Further clarification of the above challenges are provided in the sections to follow.

3.3 Data Acquisition from Web Logs

In this section, we describe the initial work carried out with respect to our approach. It focuses on data acquisition where raw web logs are converted into a format that is meaningful and easier for knowledge extraction. Data acquisition is made up of the following three tasks: Web Logs Filtration, Crawler Logs Elimination and Sessionisation. Web logs filtration involves identifying unwanted document accesses from the raw logs thereby leaving only document accesses that could be used to infer users' browsing behaviours. The crawler logs elimination task shares the same goal with the former task. It determines the document accesses that are made by automated programs called bots or crawlers and removes them, leaving only those document accesses that are associated to human users. Finally, the sessionisation task involves identifying and isolating the series of documents that was viewed by a user during his visit to the website.

Various off-the-shelf applications such as (SPSS, 2008; WebTrend, 2008) are readily available to mine web logs. However, we decided to develop one to better understand and appreciate the difficulties involved in the mining process. Furthermore, developing such an application provides a wider range of flexibility and customisation to better suit the purpose of this study as opposed to commercially available applications. As such, we have implemented an application called *LogParser* that analyses and extracts users' browsing behaviour patterns from a collection of web logs. This tool provides a general framework to process raw web logs appearing in common log format and presents the data in a form ready for the application of web usage mining techniques for knowledge discovery.

3.3.1 Web Log Filtration

Given the sheer volume and diversity of document requests in the web logs, it is important to distinguish between requests that carry vital information with regards to a user's browsing behaviour and those that do not. The web log filtration task aims to do just that by filtering out requests that are inadequate for usage mining from the web logs. For the purpose of this study, we decided to filter out all requests of the following kinds: request for graphical contents, requests for non-HTML files, unsuccessful requests and requests made by bots or crawlers.

Most requests for graphical contents and non-HTML files usually are part of an actual request made to the server to retrieve a HTML document. For example, the entirety of a HTML document could be made of text combined with graphical contents and other non-HTML files at times. Since these graphical contents and other non-HTML files need to be retrieved from the web server to properly display the document, they appear as individual requests in the web logs. However, the request that could be used to elicit information about a user is the one that retrieves the main document rather than its supporting files. Hence the need arises to eliminate such requests related to supporting role from the web logs.

Unwanted graphical and non-HTML requests are identified by referring to the retrieved file's extension in a log's request field. A log's request field records the name of the file that was retrieved and the method that was used to get it[1]. For example, a HTML document request usually has a filename that ends with a ".html", ".shtml", ".asp" or ".php" extension, while graphical contents and non-HTML file requests end with extensions such as ".jpg", ".bmp", ".gz", etc. Based on this observation, a list was created to remove requests related to graphical contents and non-HTML files. Table 3.1 shows the list of file extensions used by the *LogParser* application to remove unwanted graphical and non-

[1]Refer to figure 2.1 in chapter 2 for a complete listing of the fields that make up a request log

HTML requests from the web logs.

File Extensions	**File Type**
.gif, .jpg, .jpeg, .bmp,	Images
.au, .midi, .mpeg, .mp3, .wma, .mid, .swf	Media
.java, .js, .class, .jar	Java Files
.mdb, .dat	Data files
.gz	GNU Zip file
.ps	Post Script
.zip	Winzip file
.css	Style Sheets
.xls	Excel Spreadsheet
.clp	CLIP files
.arff	Weka files
.pac	Proxy Script
.pps	Powerpoint Files
.gs	Ghost Script

Table 3.1: Removal List

Unsuccessful requests refer to all failed attempts in retrieving and displaying a document from the web server to a user. These include both client and server side web errors which prevent access to the document being requested. One such error commonly encountered by web users' is the "404: File not found" error indicating a missing document which, could be due to relocation or deletion of the file. Unsuccessful requests do not provide much information about a user's browsing patterns, which is vital for this study and therefore is made redundant. A successful or unsuccessful request is identified by referring to a log's status field. The status field of a log contains a response code[2] returned by the server to the client indicating the status of the request. A successful document request is indicated by a response code of 200, while any other code represents a failure to retrieve the said document. The *LogParser* application takes advantage of this field to identify and remove unsuccessful requests from the web logs.

Consequently, requests made by bots or crawlers cannot be detected using straightforward methods mentioned above. The task requires a more elaborate measure to identify

[2]For a full listing of the response code refer to appendix B

and eliminate such requests. The following section will describe the difference between a user's request and of a crawler before going on to discuss the method employed to eliminate crawler requests.

3.3.2 Crawler Logs Elimination

Crawler, also referred as bot, is an automated program that browses the Web in a systematic manner. It is mostly employed by search engines to create a copy of all the visited documents, which is then indexed in order to provide fast searches. A web server does not distinguish between a request made by a crawler or a genuine user as all requests for a document are considered coming from an individual user. As such, the web logs are infused with repeated automated requests that are not made by a human user. If this data is used in the mining process it may lead to biased outcomes and therefore requests belonging to crawlers need to be eliminated from the web logs.

Eliminating requests made by crawlers are more difficult as they need to be explicitly identified. The DNS/IP Check and the Robots Exclusion Check may be the simplest form of crawler detection methods available. The DNS/IP check method involves detecting a crawler's requests by matching the originating IP address from the remote host field of a log with those of known crawlers. For this, a text file containing 199 distinct known crawler DNS/IP addresses was created. There are public databases like the Web Robots website[3] that maintains a database of known crawler DNS/IP addresses which, could be used to create the text file. By using this file the *LogParser* application is able to cross reference it with the originating IP address of each request and remove those associated to crawlers. However, note that the publicly available crawlers list is not exhaustive and it is becoming extremely difficult to keep up with the continuous evolution of the crawlers (Lourenco and Belo, 2006).

In order to detect crawlers that are not included in the list, the Robots Exclusion Check method was used in combination with the DNS/IP Check. The Robots Exclusion Check is a method based on the Robot Exclusion Standard[4]. According to the standard,

[3]http://www.robotstxt.org
[4]http://www.robotstxt.org/wc/norobots.html

crawlers are required to access the "robots.txt" exclusion policy file that resides in the web server, prior to visiting any other documents in a website. Hence any client that has made a request for the exclusion policy file can be considered as a crawler. The *LogParser* application takes advantage of this and uses the log's request field containing the name of the file retrieved to determine whether the client is a human user or a crawler. Any client's requests associated to the exclusion policy file is then eliminated from the web logs leaving only genuine users' requests.

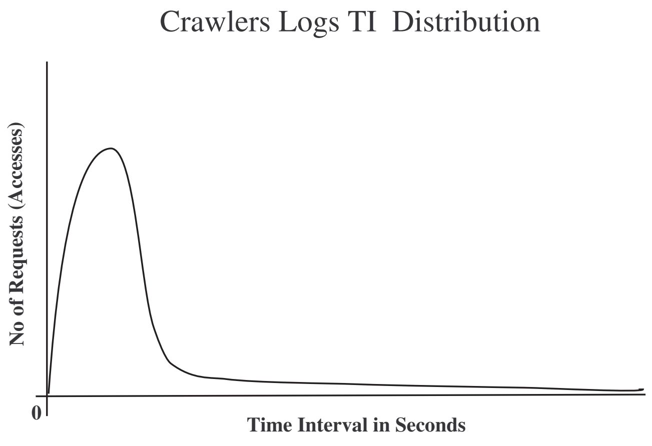


Figure 3.3: Expected Crawlers Logs TI Distribution

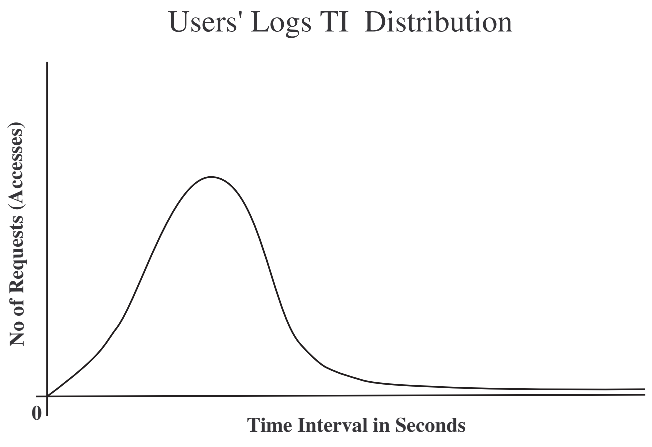


Figure 3.4: Expected Users Logs TI Distribution

Experiments were carried out to determine how well these methods performed using

3 days worth of web logs from the October 2004 dataset, containing a total of 64182 requests made by 736 distinct clients. The objective of the experiment was to find out the degree of crawlers requests still present in the users' requests and vice verse after the deployment of the combined method. One way of confirming this would be to produce a graph which, illustrates the Time Interval (TI) distribution between subsequent requests of both crawlers and users' logs found. The assumption here is that crawler being an automaton tends to perform a high number of short period visits as opposed to human users who make a high number of longer period visits. Therefore, by computing the amount of time spent for each request and plotting it against the total number of requests made within a specified time span, will allow us to gauge the performance of the combined method. Figures 3.3 and 3.4 show the expected projection of the TI distribution graphs that is to be produced for crawler's and user's logs respectively. It can be observed from figure 3.3 that we are expecting a curve with a sharp ascent early in the graph and then falls rapidly before normalising as time progresses to indicate the high number of short period visits made by crawlers. While in figure 3.4 we are expecting the curve to start off low while steadily increasing before reaching a peak and then gradually descends before flattening out representing the high number of longer period visits made by users.

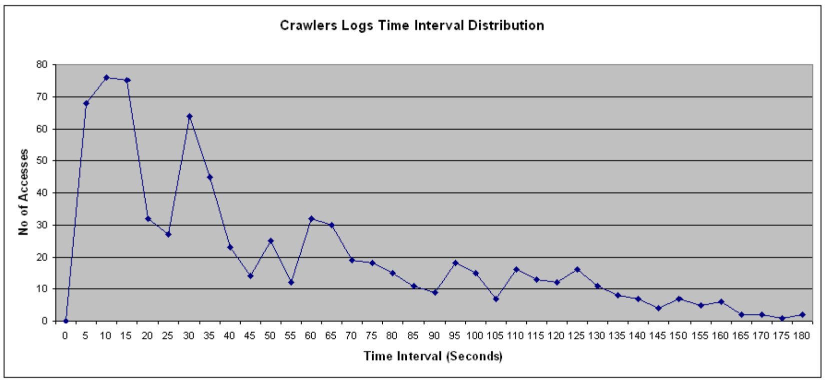


Figure 3.5: Crawlers Logs TI Distribution

Consequently, figures 3.5 and 3.6 show the actual TI distribution graphs that was obtained from the crawlers and users' logs respectively. The graphs only show requests which were made in the first 3 minutes of both crawlers and users' logs as that was the

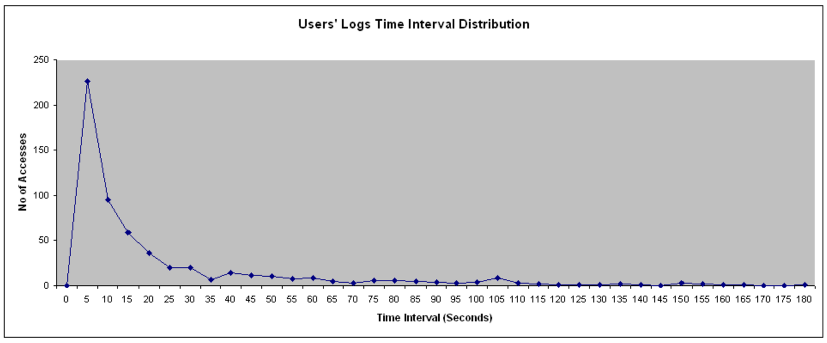


Figure 3.6: Users Logs TI Distribution

period with the most number of activities in the website. As it can be observed, the difference between the expected and actual graphs are large especially for the crawlers logs TI distribution. The outcome for the crawlers logs TI distribution was completely unexpected as it did not match our prediction of a crawler's browsing behaviour. The TI distribution pattern illustrated in figure 3.5 is far too erratic to belong to either crawlers or users. As such, further investigation was carried out to explain the reason behind this occurrence. This investigation involved in randomly selecting 3 known crawlers from the 3 day dataset and plotting a TI distribution graph for every request made by them. Figure 3.7 shows the TI patterns obtained from the 3 randomly selected crawlers. The first crawler's TI pattern matches the expected TI graph in figure 3.3 to a certain degree but it does not normalise as projected. While the second crawler's TI pattern resembles more of the user's expected TI graph though in actuality it is a known crawler. Finally, the third crawler's TI pattern appears to be more of a mixture of both expected TI graphs. Based on figure 3.7 it can be concluded that crawlers are indeed continually evolving and thus making it difficult to detect them unless they are explicitly identified[5].

The outcome for the users' logs TI distribution in figure 3.6 was also unexpected as it was more similar to the expected crawler's TI pattern rather than the user's TI pattern. The sharp increase in the number of requests at the beginning of the graph followed by a gradual decline before normalising matches the predicted behaviour of crawlers. A close inspection of the dataset revealed that these requests may be a result of a set of fast

[5]Explicit identification involves detecting a crawler by means of DNS/IP address cross referencing or access to the robots.txt exclusion policy file.

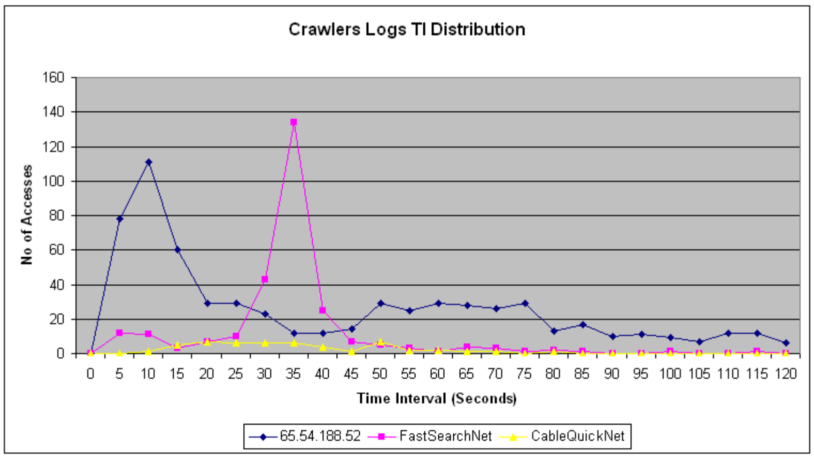


Figure 3.7: TI Distribution of 3 Individual Crawlers

clicks performed by users before reaching their document of interest where they spend a longer reading time. This phenomenon could be regarded as a common practice among experienced users who are accustomed to the website's navigational structure due to their repeated visits to it. Regardless of the unexpected outcomes, it could be concluded that the combined method performs reasonably in distinguishing between known crawlers requests and users' requests.

3.3.3 Sessionisation

Sessionisation is the final task in the process of acquiring data from web logs. The sessionisation task involves in identifying and reconstructing the navigational route or routes taken by a user during his visit to the website. In this study, a session can be described as a group of documents requested by a user while browsing, which makes up his navigational route through the website. A user's visit to a website may consists of one or more number of sessions and the same user could also make multiple number of visits in a day. In order to simplify the process of identifying a user's session, we decided to consider all visits made by the user in a single day as one individual visit. Furthermore, we've realised that the proportion of work involved in identifying users' sessions can be of a full fledged study in itself with respect to the two major issues in identifying a user's session introduced in section 2.1.2. In our case, it is extremely difficult to trace the occurrences of document backtracking caused by local caching since our source of data lacks the information required to do so. In order to trace document backtracking, it would require the resetting of the school's web servers to implement the use of referrer logs and recollect the source data to be used for this study. The excessive amount of time and work required coupled with the University's regulation on such drastic changes to the school's web infrastructure made it impossible to employ such a method. Therefore, we decided to use the existing source data as it is even though it does not capture the document backtracking information.

Similarly the issue of proxy caching requires the use of additional data such as the proxy logs to identify multiple users assigned with the same DNS/IP address. However, the use of such proxy logs only allows us to identify the users that go through the school's proxy server rather than external proxy servers. Furthermore, the internal proxy logs are neither tracked nor kept as long as the school's web logs. Therefore we decided to minimise our computational overhead by removing all document requests associated to the school's proxy server. The school's proxy server handles all requests for document from the internal users and assigns the same originating DNS/IP address, i.e., roadkill as identifier for the requests. The LogParser application takes advantage of this to distinguish and remove requests belonging to the school's proxy server from the web logs. Note that the same method could not be applied to identify requests associated to external proxy servers as the information (i.e., external proxy logs or external proxy DNS/IP) required to do so was not available. Although this does not completely eliminate the problem of proxy caching, it does reduce the proportion of proxy related requests in the web logs and any inconsistent users' sessions that could be generated.

In order to identify users' sessions from the remaining web logs a method based on Goker, He and Harper's (2002) web session identification approach was employed. Goker et al.'s (2002) approach identifies document requests belonging to a session through close proximity in browsing time and additional contextual information (i.e. document topics). However, in our study, due to the absence of additional contextual information in the web logs we decided to use the time difference between subsequent document requests as a way of identifying users sessions. According to (Goker and He, 2000a; Goker and He, 2000b) there exists a correlation between the time difference (i.e., time interval) of subsequent requests and session shift[6] where the longer the time interval is between subsequent requests, the higher the likelihood that there is a session shift.

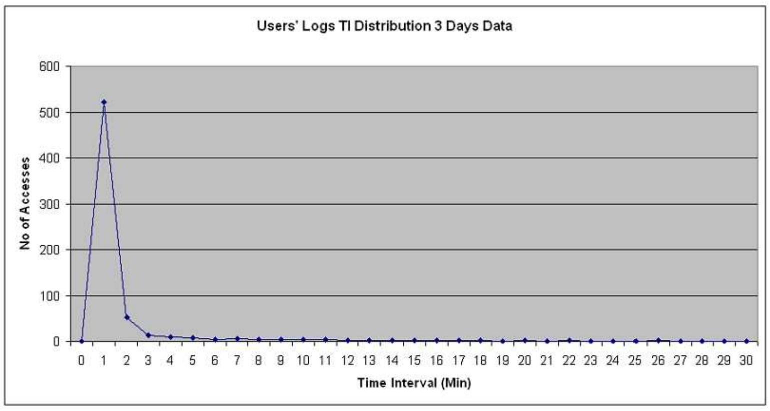
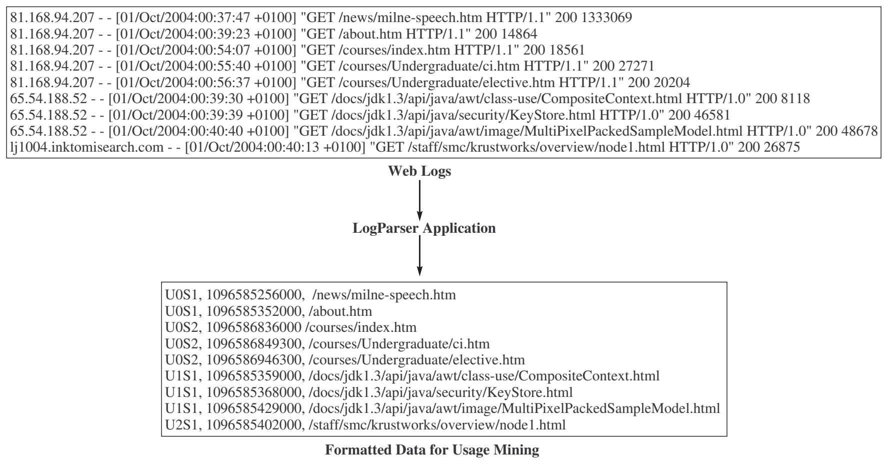


Figure 3.8: Users Requests Time Interval Distribution for Determining Threshold

In our study, to begin with, each unique originating DNS/IP address of a document request was assumed to represent an individual user. A predefined time threshold is then used to indicate the end of a user's session whenever the time interval between subsequent requests exceeds the threshold. The threshold was determined by producing a graph which illustrates the time interval distribution between subsequent requests of web logs. The graph as shown in figure 3.8 was generated using 3 days worth of web logs containing a total of 64182 requests made by 736 distinct clients. The graph only features requests with time interval up to 30 minutes because there were no significant increase in the number of requests beyond it. As it can be observed, most of the users' activities happens within the first 3 minutes of the time interval after which, the number of users' requests steadily decreases. There are still small traces of activities going on until the 12th minute although the number of requests are not significant enough. Nevertheless, based on the graph it may be concluded that the longest time a user might spend browsing the website is at most 12 minutes after which the number of users' requests become extremely low or non-existent. Therefore the time threshold to identify the end of a user's sessions was set at 12 minutes.

The LogParser application uses the predefined time threshold to determine users' re-

[6]Session shift refers to the beginning of a new session happening between two subsequent requests from the same user



```
81.168.94.207 - - [01/Oct/2004:00:37:47 +0100] "GET /news/milne-speech.htm HTTP/1.1" 200 1333069
81.168.94.207 - - [01/Oct/2004:00:39:23 +0100] "GET /about.htm HTTP/1.1" 200 14864
81.168.94.207 - - [01/Oct/2004:00:54:07 +0100] "GET /courses/index.htm HTTP/1.1" 200 18561
81.168.94.207 - - [01/Oct/2004:00:55:40 +0100] "GET /courses/Undergraduate/ci.htm HTTP/1.1" 200 27271
81.168.94.207 - - [01/Oct/2004:00:56:37 +0100] "GET /courses/Undergraduate/elective.htm HTTP/1.1" 200 20204
65.54.188.52 - - [01/Oct/2004:00:39:30 +0100] "GET /docs/jdk1.3/api/java/awt/class-use/CompositeContext.html HTTP/1.0" 200 8118
65.54.188.52 - - [01/Oct/2004:00:39:39 +0100] "GET /docs/jdk1.3/api/java/security/KeyStore.html HTTP/1.0" 200 46581
65.54.188.52 - - [01/Oct/2004:00:40:40 +0100] "GET /docs/jdk1.3/api/java/awt/image/MultiPixelPackedSampleModel.html HTTP/1.0" 200 48678
lj1004.inktomisearch.com - - [01/Oct/2004:00:40:13 +0100] "GET /staff/smc/krustworks/overview/node1.html HTTP/1.0" 200 26875
```

```
U0S1, 1096585256000,  /news/milne-speech.htm
U0S1, 1096585352000, /about.htm
U0S2, 1096586836000 /courses/index.htm
U0S2, 1096586849300, /courses/Undergraduate/ci.htm
U0S2, 1096586946300, /courses/Undergraduate/elective.htm
U1S1, 1096585359000, /docs/jdk1.3/api/java/awt/class-use/CompositeContext.html
U1S1, 1096585368000, /docs/jdk1.3/api/java/security/KeyStore.html
U1S1, 1096585429000, /docs/jdk1.3/api/java/awt/image/MultiPixelPackedSampleModel.html
U2S1, 1096585402000, /staff/smc/krustworks/overview/node1.html
```

Figure 3.9: Session Identification and Formatted Data

quests that belong together and groups them in a single session, following which the web logs are rewritten in a format ready for usage mining to a comma separated text file (CSV). Furthermore, due to privacy reasons each request's originating DNS/IP address in a session is anonymised to a unique client/session ID before it is rewritten to the CSV file. Figure 3.9 shows the process of identifying users' sessions and an extract of the formatted data generated for usage mining. Observe how each unique DNS/IP address in the users' requests is replaced by an anonymous identifier such as U1, U2 and U3 indicating users 1 to 3. Then for each user, the requests belonging to a session is grouped together based on the time threshold and assigned a session identifier corresponding to the number of sessions found such as S1, S2, etc. Once all users' sessions have been identified the data is then rewritten in the following format for usage mining: User/SessionID, Date and Time of request in miliseconds, Filename of requested document.

3.4 Chapter Summary

In this chapter, we have described that a complex website design impedes the users effort in navigating the website and identified a drawback in the existing shortcutting methods used to solve the above mentioned problem. With respect to this, we have introduced our approach to adaptive websites by identifying and suggesting wayposts as navigational shortcuts that could be used to improve a website's organisation. We also described the advantage of our approach which gives significance to the intermediate documents appearing in a path as opposed to the existing methods. Furthermore, we pointed out that web logs are rich in information on users' browsing patterns and website usage. However, prior to harnessing such information, raw web logs need to be filtered to remove users' requests which do not contribute towards usage mining. In line with this, we have identified the three tasks that performs data acquisition from web logs. The process involved in each task were discussed in detail and justification of the methods employed to achieve them were also provided, following which the data acquired from the web logs is then represented in a format ready for knowledge discovery.

Chapter 4

Discovering Knowledge From Data

This chapter describes the work carried out with respect to eliciting knowledge from web logs. The work consists of the following tasks: identifying a set of target documents, extracting users' navigational paths corresponding to the target documents, grouping similar paths together and identifying wayposts for shortcut generation. The chapter discusses the details of these tasks and elaborates on the methods employed to carry them out. Following that the chapter gives an account of the evaluation performed on our approach.

The chapter is organised as follows: Section 4.1 provides a brief description of target documents and users' paths with sections 4.1.1 and 4.1.2 discussing the methods employed to identify target documents and discover navigational paths respectively. Sections 4.2 presents the approach used to group similar paths together and gives examples of clusters generated. In section 4.3, the means of identifying wayposts from the clusters generated is described and a few shortcuts generated from the wayposts found are presented. Finally, section 4.4 presents an evaluation of our approach.

4.1 Target Documents and Paths

A user's activity in a website is driven by the need for information corresponding to a goal in his mind. The collection of documents visited by the user in search of the information that could satisfy his current need is referred to as a user's browsing session. The end of a user's browsing session could mean either that the user has found the information being searched for or the user has failed to do so. Documents in a website that contain the information a user is searching for with regards to his current needs are known as target documents. The series of documents that were visited by the user to reach these target documents are known as the user's path.

Our approach identifies wayposts from frequently travelled users' paths. In order to do so, each user's navigational path beginning at an initial document and terminating at a target document needs to be discovered. The following section describes two different methods that were used to identify potential target documents from the data acquired of web logs and discusses their effectiveness based on the target documents found. Section 4.1.2 describes the process of extracting paths from users' session based on the identified target documents.

4.1.1 Identifying Target Documents

The task of identifying target documents correctly is critical for path discovery. The issue is to find a reasonable way of distinguishing target documents from the collection of documents in a website. Here, we describe and evaluate two different methods that were used in our work to identify a set of target documents from the School of Computing's website, namely the end document method and browsing time method. The end document method relies on users' visit frequency information, while the browsing time method relies on the documents visit time information, both of which are obtained from the web logs, to determine if a document is qualified to be a target document.

End Document Method

As mentioned earlier, a user's session consists of a collection of documents visited by the user in search of a target document. The termination of a user's session could mean either the user has reached the goal or has not. However, repeated visits to a particular document by users could indicate a high level of interest among them in the document and therefore increases its likelihood of being a target document (Lieberman, 1995). Similarly, the end document method looks at the number of times a particular document appears at the end of a user's session. The repeated occurrence of a document as the final document visited in multiple users' sessions indicates that it may be of interest to the users and therefore it is likely to be a target document. In order to evaluate the effectiveness of this approach in identifying potential target documents, two separate experiments were carried out. These experiments were carried out using a data set consisting of 395058 document accesses made by 2233 clients to the School's web server from the 1^{st} to 7^{th} October 2004.

The first experiment involved identifying all unique documents that appear at the end of each user's session. Then for each of these documents the number of times it appears as the last document visited in users' sessions is computed. However, we realise that this could result in countless number of documents to be identified as target documents. Therefore a threshold was introduced to filter out documents with low occurrence frequency as an end document in users' sessions. Table 4.1 shows the documents that were identified as target documents based on their corresponding frequency value as end documents in users' sessions. In the table, End Document Frequency (EDF) refers to the number of times the document occurs as the last document visited in users' sessions.

As it can be seen from table 4.1, most of the documents identified as target documents belonged to staff webpages and courses offered webpages of the School's website. Staff webpages are owned and maintained by individual people employed in the School, mainly lecturers. These documents contain a wide range of information from research papers to teaching materials associated to the particular staff's role in the School. Documents such as these are quite popular among current students who have enrolled in the School, as they are their main source of information to keep track of their course module delivery and obtain lecture notes posted by their lecturers. The courses offered webpages provide information on specific courses available in the School and are critical point of attraction for potential future students. These documents contain information such as course name, course duration, entry requirements, modules that will be taught and course fee among others. Similarly, "support.html" and "contacts.html" having EDF values of 28 and 26 respectively are also common documents in the School's website that users' tend to visit frequently. The "support.html" document contains details of the School's IT support

Page ID	EDF	Page Name
P1	12	/staff/dc/hcow/menu.html
P2	16	/staff/dc/hcow/cow_noframe.html
P5	55	/
P24	27	/welcome.ht
P26	5	/about.htm
P27	9	/courses/index.htm
P29	5	/courses/Undergraduate/elective.htm
P36	18	/staff/dc/hcow/links.html
P39	29	/courses/postgradindex.htm
P46	5	/e-softtech/
P50	5	/e-softtech/fees.htm
P59	4	/staff/khui/teaching/cm3016/
P65	6	/staff/rab/Music/musichome.html
P71	28	/support.htm
P74	26	/contacts.htm
P90	21	/staff/dc/hcow/discog.html
P91	4	/staff/dc/hcow/articles.html
P107	5	/courses/Undergraduate/cga.htm
P108	4	/school.htm
P109	7	/courses/Undergraduate/cim.htm
P111	11	/support/laptopsforstudents/index.htm
P113	5	/courses/Undergraduate/cbe.htm
P114	8	/courses/Undergraduate/cs.htm
P128	8	/courses/Postgraduate/mcie.htm
P132	9	/courses/Postgraduate/mcst.htm
P266	4	/staff/nw/NwLk.html
P273	4	/staff/sy/
P291	4	/staff/ses/
P301	5	/staff/
P306	21	/staff/nc/pages-of-nav-lecture-notes/CM3006.htm
P313	8	/staff/chb/cm4007/cm4007.html
P337	10	/local/courses/projects/guidelines.html
P357	10	/staff/dc/hcow/news.html
P358	5	/staff/dc/hcow/whats_new.html
P385	9	/staff/dc/hcow/musicians.html
P386	5	/staff/dc/hcow/rel_bands.html
P402	5	/staff/db/Seminars/HoPL/sld001.htm

Table 4.1: Target Documents Identified Based on EDF from The 01 to 07 Oct Web Logs

team, while the “contacts.html” document lists names and contact details of people to who general or specific enquiries about the School and courses may be directed to. Due to the critical nature of the information contained in all the documents mentioned above, it is reasonable for them to be identified as potential target documents in the website. However, we noticed that the document with the highest EDF value of 55 is actually the School’s index document. This was unexpected, as we could not find a scenario where a user would choose the index document of a website as his target document. One possible explanation could be that users failing to find the information they were searching for, attempted to restart their browsing session from the index document but decided otherwise for some unknown reason.

Although the above experiment provided us with reasonable results, we realise that relying purely on the EDF value of a document to indicate its candidacy as a target document is an overestimation of the importance of the document to a user. This can be solved by taking into account the number of incoming visits (IV)[1] to a document and comparing it against the document's EDF. By doing so, it allows for a more reasonable estimation of determining the importance of a document's contribution towards a user's browsing need and thereafter its candidacy as a target document. In this refined version of the above mentioned approach, we consider documents with a high number of EDF value and incoming visits are more likely to be a target document. Based on this, a document needs to satisfy two important conditions to be considered as a target document and the conditions are as listed below:

$$\frac{EDF}{IV} \geq 0.5 \tag{4.1}$$

$$EDF \leq IV \tag{4.2}$$

The first condition, as shown in equation 4.1, states that the ratio of a document's EDF value to its number of incoming visits should be more than or equal to 0.5. What is meant by this is, if a document's EDF value (i.e., the number of times it appears as the last document visited in users' sessions) is at least half the number of its incoming visits, then it can be considered as a target document. The second condition in equation 4.2 defines that the document's EDF value must be less then or equal to its number of

[1]Number of times a document has been visited.

incoming visits. Essentially it means, the number of times an end document can appear at the termination of users' sessions has to be less than or equal to the number of visits it receives. A second experiment was carried out using the same data set as in the first experiment, in order to evaluate the effectiveness of this refined approach in identifying potential target documents. The results of the experiment are as shown in tables 4.2 and 4.3.

Table 4.2 shows the EDF, IV and EDV/IV values computed for the same set of documents that were identified earlier in the first experiment. It also includes the compliance of these documents with respect to the two conditions introduced above. As it can be seen from table 4.2, several documents that were identified as target documents in the first experiment now no longer qualifies as one. These documents have failed to comply with the first condition which requires their EDF values to be at least half of their IV values. As it can be observed from table 4.2, these document have a higher number of incoming visits but their occurrences as an end document in users' sessions is not high enough for them to be considered as target documents. In addition, note how the School's index document (P5) did not get selected as a target document using this refined method. This indicates that the earlier method based solely on the EDF values indeed overestimated the importance of these documents to users. Nevertheless, there are a few documents which managed to satisfy the two conditions mentioned above and were identified as target documents. These documents are shown separately in table 4.3. Note that these documents comprise of the similar group of documents that were highlighted in the first experiment namely the staff webpages, courses offered webpages, "support.html" and "contact.html". These documents have a high EDF value in proportion to the number of visits they receive. This emphasises that they are of interest to users and therefore, are more suitable as target documents. From this outcome, we can conclude that this refined method based on the EDF/IV value provides an improved estimate if not precise, of a document's likelihood to be a target document.

Page ID	EDF	IV	EDF/IV	1st Condition	2nd Condition
P1	12	102	0.12	No	Yes
P2	16	112	0.14	No	Yes
P5	55	129	0.43	No	Yes
P24	27	138	0.02	No	Yes
P26	5	30	0.17	No	Yes
P27	9	142	0.06	No	Yes
P29	5	11	0.05	No	Yes
P36	18	31	0.58	Yes	Yes
P39	29	67	0.43	No	Yes
P46	5	17	0.29	No	Yes
P50	5	20	0.25	No	Yes
P59	4	8	0.50	Yes	Yes
P65	6	14	0.43	No	Yes
P71	28	53	0.53	Yes	Yes
P74	26	43	0.60	Yes	Yes
P90	21	54	0.39	No	Yes
P91	4	20	0.20	No	Yes
P107	5	15	0.33	No	Yes
P108	4	21	0.19	No	Yes
P109	7	22	0.32	No	Yes
P111	11	17	0.65	Yes	Yes
P113	5	11	0.15	No	Yes
P114	8	28	0.29	No	Yes
P128	8	12	0.67	Yes	Yes
P132	9	20	0.45	No	Yes
P266	4	4	1.00	Yes	Yes
P273	4	6	0.67	Yes	Yes
P291	4	7	0.57	Yes	Yes
P301	5	17	0.29	No	Yes
P306	21	25	0.84	Yes	Yes
P313	8	12	0.67	Yes	Yes
P337	10	16	0.63	Yes	Yes
P357	10	29	0.34	No	Yes
P358	5	21	0.24	No	Yes
P385	9	44	0.20	No	Yes
P386	5	26	0.19	No	Yes
P402	5	17	0.29	No	Yes

Table 4.2: EDF/IV Value Computed for End Documents Found from 01 to 07 Oct Web Logs

Browsing Time Method

One limitation of the end document approach is that it only considers the documents visited at the end of users sessions as potential candidate for target documents. It does not take into consideration the possibility that the intermediate documents in each user's session could also be a target document. In order to address this, we looked at another

Page ID	EDF	IV	EDF/IV	Page Name
P36	18	31	0.58	/staff/dc/hcow/links.html
P59	4	8	0.50	/staff/khui/teaching/cm3016/
P71	28	53	0.53	/support.htm
P74	26	43	0.60	/contacts.htm
P111	11	17	0.65	/support/laptopsforstudents/index.htm
P128	8	12	0.67	/courses/Postgraduate/mcie.htm
P266	4	4	1.00	/staff/nw/NwLk.html
P273	4	6	0.67	/staff/sy/
P291	4	7	0.57	/staff/ses/
P306	21	25	0.84	/staff/nc/pages-of-nav-lecture-notes/CM3006.htm
P313	8	12	0.67	/staff/chb/cm4007/cm4007.html
P337	10	16	0.63	/local/courses/projects/guidelines.html

Table 4.3: Target Documents Identified Based on EDF/IV Values Computed

method of identifying target documents from the School's website. The browsing time method is commonly used as a way of identifying target documents within a website. The method is based on the idea that users tend to spend a longer time on interesting documents rather than uninteresting documents. Therefore, by setting a time threshold we can determine whether a document is a target document or not (Godoy and Amandi, 2006). However, the issue here is finding a suitable threshold value that can be used to correctly identify target documents. In order to decide on a suitable threshold value, we reused the users' logs TI distribution graph plotted in section 3.3.2 to find a point in the graph that could indicate the average time most users would spend visiting a document. The graph is reproduced below in figure 4.1 with a reference to the point selected from the graph indicating the value selected for the threshold.

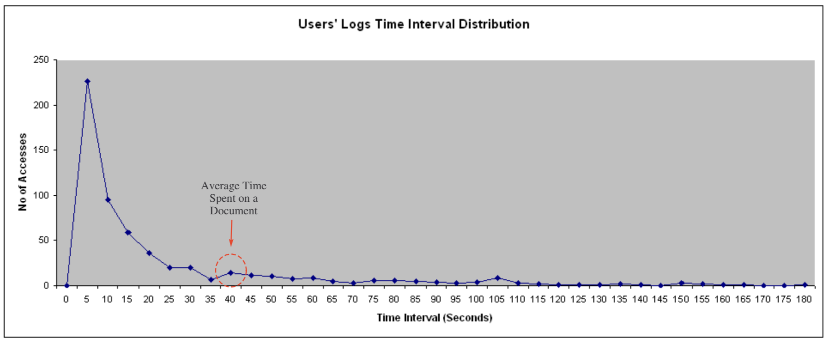
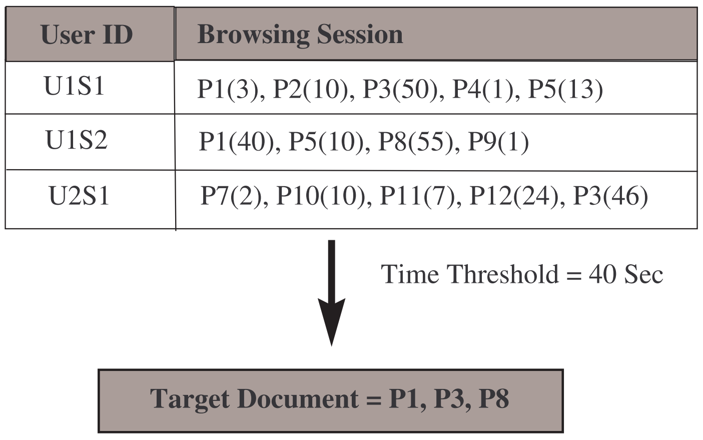


Figure 4.1: Finding Time Threshold from Users Logs TI Distribution

The graph shown in figure 4.1 is obtained by computing the time spent for each document request in users' logs and plotting it against the total number of requests made within a specified time span. As noted in section 3.3.2, the graph only show requests which were made in the first 3 minutes of the users' logs as that was the period with the most number of activities in the website. The sharp increase and decrease of requests in the beginning of the graph, was referenced to as possibly a set of fast clicks made by users before reaching their document of interest where they spend a longer reading time. We noted that this phenomenon could be a common practice among experienced users who are accustomed to the website's navigational structure. Based on this we deduced that there should be a second increase in the graph which, could represent the average time most users would spend visiting a document. It can be observed from the graph that there is a point where the decrease in the number of users requests comes to a temporary halt and shows sign of very small increase before decreasing further down and fluctuating thereafter. We believe that this second peak in the graph, where there is a small increase in the number of users requests, could be an indication of users reaching their target documents after a series of fast clicks. Therefore, based on this observation the threshold value to identify a target document was set to 40 seconds.

Figure 4.2 shows how target documents are identified from users' session using the time threshold. The user id column indicates the anonymised id assigned to each user and his respective browsing sessions. The browsing session column shows the collection of documents visited by each user during his browsing session where the documents are represented using anonymised page id. The number appearing inside "()" represents the time spend by the user in the corresponding document. As shown in figure 4.2, all unique documents with users browsing time exceeding the time threshold are identified as target document.

In order to evaluate the effectiveness of this method in identifying target documents, an experiment was carried out using the same data set that was employed in the end document method. The data set contained a total of 395058 document accesses made by 2233 clients to the School's web server from the 1st to 7th October 2004. The set of target



User ID	Browsing Session
U1S1	P1(3), P2(10), P3(50), P4(1), P5(13)
U1S2	P1(40), P5(10), P8(55), P9(1)
U2S1	P7(2), P10(10), P11(7), P12(24), P3(46)

Figure 4.2: Finding Target Documents Based on Time Threshold from Users Sessions

documents that were identified from the data set using the browsing time method can be found in appendix C[2]. Based on the our observation of the obtained result, it was found that this method generated far too many target documents which could affect the later task of path identification. An examination of the set of target documents found revealed that the majority of documents found by this method are actually short term target documents rather than long term ones. Short term target documents are documents in which a user spends enough time for it to be identified as a target document but it does not contain the information necessary to satisfy the user's original goal. For example, a student might be browsing for a document containing lecture notes on Java programming in the website when, the student notices a link to the graduate recruitment fair and decides to take a short detour before resuming and finding the document containing the Java lecture notes. Although the student spent enough time in both these documents for them to be identified as target documents, it is clear that only one of the document contribute toward his original goal. Therefore, short term target documents can be associated to a sudden shift of interest in a user's mind while the user is browsing towards the original goal. One reason for such short term target documents to be picked up along with long term target documents, could be due to the threshold value selected being not a good estimate of a document's candidacy as a target document. Furthermore, short term target documents

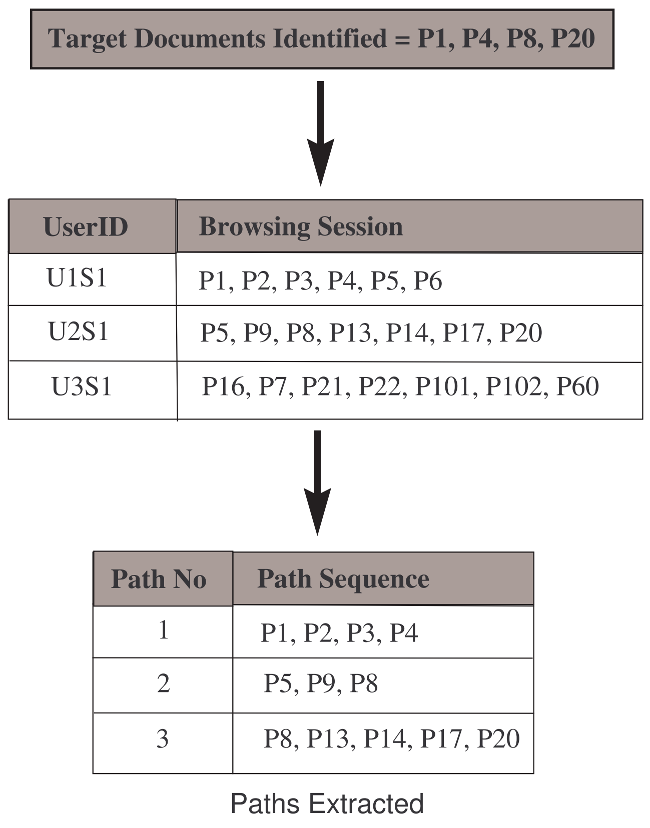
[2]Note: Only a small sample of the target documents found will be listed in the appendix due to the large number of target documents that was found.

are inadequate for the later task of path extraction as it causes a large number of short paths to be generated from the users' session. On the other hand, despite its limitation the end document method provides a more reasonable estimate of a documents candidacy as a target document. Therefore we decided to use the end document method as the mean of identifying a set of target documents from users' sessions in our work, which can then be used for path extraction.

4.1.2 Extracting Paths

As defined earlier in section 3.1, a path can be thought as a user's journey from an initial document to a target document in the website. The sequence of documents visited along this journey indicate the navigational route taken by a user to reach his intended target document. The task of path extraction involves identifying such users' paths by breaking down each user's session into a collection of navigational route terminating at a target document. In order to do this, it relies on the set of target documents identified earlier in the target document identification task. The number of paths that can be extracted from a user's session is predominantly determined by the number of target documents present in it. Therefore, a user's session could consists of a path, multiple paths or no paths at all with respect to the number of target documents found in it.

Figure 4.3 shows how paths are extracted from users' sessions given a set of target documents. Each user's session is represented by a unique id followed by the collection of documents visited by him in his browsing session. In the figure, each "P" followed by a number represents a document that the user visited in his browsing session. Note how although there are two target documents in U1S1, only one path is extracted from it. This is because "P1" happens to be the initial document from which the user started his browsing session, therefore this overwrites its status as a target document. On the other hand, note that session U2S1 is split into two different paths. In such cases, where multiple target documents are found within a user's browsing session, the target document of the first path becomes the initial document for the next path and it continues likewise for the other target documents found in the user's session. Finally, session U3S1 represents the



Target Documents Identified = P1, P4, P8, P20

UserID	Browsing Session
U1S1	P1, P2, P3, P4, P5, P6
U2S1	P5, P9, P8, P13, P14, P17, P20
U3S1	P16, P7, P21, P22, P101, P102, P60

Path No	Path Sequence
1	P1, P2, P3, P4
2	P5, P9, P8
3	P8, P13, P14, P17, P20

Paths Extracted

Figure 4.3: Extracting Paths from Users' Sessions Based on A Set of Target Documents

case where no paths could be extracted due to the absence of a target document in the user's browsing session.

We note that the paths obtained through this extraction method varies in length and may not always represent ideal cases as shown in figure 4.3. Some of the extracted paths may only have 2 documents in them, which is the minimum length a path could consists of. However, such paths are too short to be considered for wayposts identification and therefore are filtered out from the collection of paths found prior to the next task of finding and grouping similar paths together. From this point onward in this thesis, all experiments were carried out using a 2 week data set consisting of 862493 document accesses made by 4071 clients to the School's web server from the 1^{st} to 14^{th} October 2004. Users' sessions were discovered from the 2 week data set and a set of target documents were identified from them. These target documents were then subsequently used to extract paths from each user's session discovered. A total of 451 paths leading to 63 different target documents were found and a small sample from this collection of paths is listed in appendix D.

4.2 Finding and Grouping Similar Paths

In a website, there may be several different ways in which the users could reach a specific target document. This suggests that there could be a multiple number of navigational paths taken by the users, which leads to the same target document. If there is such a collection of users' paths leading to the same target document, then it is reasonable to assume that they may share some common documents (i.e., potential candidate wayposts) between them. We believe that these documents may contain information with regards to a user's need that leads him to the target document. As such, it may be worthwhile providing shorcuts between these wayposts, which could assist the users in reaching their target documents. In order to identify such wayposts there is a need to group users' paths which are similar together.

Clustering is a data mining technique used to group a set of items that share similar characteristic together. It is commonly used to discover clusters of document with similar content or identify groups of users with similar browsing behaviours in a website. In our work, we will be using it to discover clusters of similar users' paths from which wayposts could be identified from. There are various clustering algorithms available but for the purpose of our work we decided to develop an implementation of the K-means algorithm to cluster the users' paths. The reason for this selection is that the K-means algorithm offers a simple and fast way of clustering similar items together. Moreover, the algorithm is widely used and it generates reasonably good clusters of similar items, given a sensible distance measure.

A key issue in implementing the K-means clustering algorithm is the need for a suitable distance measure which can determine the similarity between two paths. We decided to use the number of co-occurring documents between two paths as a measure of their similarity. With respect to this, we represent each user's path as a set, with its members being the sequence of documents visited by the user in the path. For example, a typical set representation of a user's path would be as follows: $A = \{P1, P2, P3, P4, P5\}$, where the alphabet A represents the path id, while the items within the "{ }" denote the documents visited by the user in this navigational path. This allows us to define the distance measure between two paths using the set equation below:

$$Distance(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.3}$$

The distance measure presented in equation 4.3, determines the distance between paths A and B, by computing the ratio of the number of overlapping items between set A and B to the total number of unique items in both sets. This gives us a distance value between 0 and 1 which is used to describe how similar the paths are. A distance value of 0 indicates that the two paths are different, while a distance value of 1 indicates that they are the same. One limitation of this distance measure is that it does not take into account the sequence in which co-occurring documents between two paths are visited. Nevertheless, it provides a reasonable way of determining the distance between two paths. On the other hand, we realised grouping paths based on this measure could generate clusters containing mixed users' paths (paths that are similar but leading to different target documents) and such clusters are unsuitable for the later task of wayposts identification. Therefore the collection of users' paths are grouped according to their target document and each of the resulting group of paths are then clustered using the implemented K-means algorithm.

The K-means algorithm works by selecting k number of random instances from a collection of instances and uses them as the initial seeds for clustering. Following that, it takes each instance in the collection and computes its distance with the k number of instances selected earlier and assigns it to the nearest k cluster. Then the algorithm selects the mean of each cluster as the new k instances and recomputes the distance for each instance in the collection with the new k instances, before reassigning it to the nearest k cluster. This process is repeated until the k number of instances achieve some sort of convergence. The following is the K-means algorithm implemented in our work to find clusters of similar paths:

For each Group of Paths `G`*, found from the collection of paths* `P`*:*

1. *Select k number of random paths from* `G` *as initial seeds.*

2. *Compute the distance of each path in group* `G` *with the k number of paths selected.*

3. *Assign each path in group* `G` *to the nearest k cluster* `C` *and compute mean of each cluster* `C`*.*

4. *Set mean of each cluster* `C` *as the new k paths and repeat steps 2 to 4, until k paths no longer changes.*

The above algorithm was used to find clusters of similar paths from the collection of paths discovered earlier in section 4.1.2. Note that the value for k was determined experimentally based on the size of each group of paths (i.e., number of paths in them) found from the collection. The higher the value for k is, the more number of groups with less than k paths in them are dropped out from clustering. Therefore, we set the value for k to 3, in order to increase the number of groups of paths that can be included for clustering. This also indirectly maximises the number of clusters which can later be used for the task of identifying wayposts. A total of 36 groups of paths, containing 409 paths were deemed appropriate for clustering by the algorithm, from the collection of 451 paths leading to 63 different target documents found earlier in section 4.1.2. Clusters of similar paths were then generated for each of these 36 groups of paths using our k-means algorithm. Appendix E, lists a small sample of clusters that were generated from 6 different groups out of the 36 groups of paths that were used for clustering. The clusters shown in appendix E, demonstrates that the distance measure described above is reasonably effective in identifying similar paths, allowing our k-means algorithm to generate clusters which, can be used for wayposts identification.

4.3 Identifying Wayposts and Generating Shortcuts

Different starting locations result in different roads (i.e, routes) to be taken to a particular destination. Nevertheless there are certain landmarks on roads which all drivers' travelling to the same destination passes through. Similarly, in a website regardless of the different paths that may be taken by the users to reach a particular target document, there are certain documents which could commonly be visited by all of them en-route to the target document. The repeated occurrences of such documents in multiple users' paths may correspond to the users interest and precedence in them as opposed to the other documents visited in the users' paths. Therefore, these documents are likely to be wayposts that could lead users to their target documents, similar to landmarks on roads. Providing a link (i.e., shortcut) between these potential wayposts could assist users by reducing the number of clicks they have to make while browsing, pointing them in the right direction towards a specific target document. In addition, by providing such a link, any information with regards to a user's need which may be contained in these wayposts are also preserved.

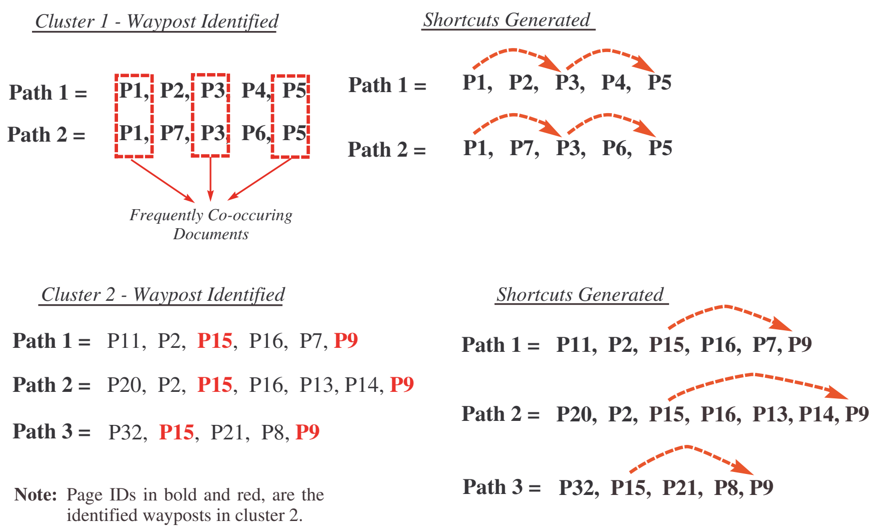

Cluster 1 - Waypost Identified
Path 1 = P1, P2, P3 P4, P5
Path 2 = P1, P7, P3 P6, P5
Frequently Co-occuring Documents
Shortcuts Generated
Path 1 = P1, P2, P3, P4, P5
Path 2 = P1, P7, P3, P6, P5
Cluster 2 - Waypost Identified
Path 1 = P11, P2, P15, P16, P7, P9
Path 2 = P20, P2, P15, P16, P13, P14, P9
Path 3 = P32, P15, P21, P8, P9
Note: Page IDs in bold and red, are the identified wayposts in cluster 2.
Shortcuts Generated
Path 1 = P11, P2, P15, P16, P7, P9
Path 2 = P20, P2, P15, P16, P13, P14, P9
Path 3 = P32, P15, P21, P8, P9

Figure 4.4: Identifying Wayposts and Generating Shortcuts

Figure 4.4 shows how wayposts are identified from clusters of similar paths and the shortcuts that could be generated from them. Cluster 1 in figure 4.4 shows two paths of the same length having three documents commonly visited between them. These documents are identified as wayposts and shortcuts are suggested from them. Of course such cases would be ideal for the task of identifying wayposts and generating shortcuts. However, not all clusters will contain ideal paths such as those. Cluster 2 represents a more realistic case, which contains three paths of different lengths and initial documents. Nevertheless the three paths still shared some common documents among them which were identified as wayposts and suggested for shortcutting. Wayposts that could be identified from such clusters are very few. Furthermore, there could even be cases where no wayposts can be generated from a cluster of similar paths, as the target document is the only document commonly shared between all the paths of the cluster. Clusters such as that suggest there is a lack in the numbers of similar user's path leading to a particular target document.

One limitation of the above approach used to identify wayposts is that it cannot be applied to clusters which consist of only one path in them. Therefore such singleton clusters are excluded from the waypost identification and shortcut generation task. Table 4.4 shows the set of wayposts identified and shortcuts suggested from the clusters of each group of paths shown in appendix E.

Group	**Cluster No**	**Wayposts Identified**	**Shortcuts Generated**
1	1	None	None
	2	P5, P27, P29	P5-P27, P27-P29
	3	P139, P29	P139-P29
2	1	P55, P95	P55-P95
	2	None	None
	3	None	None
3	1	None	None
	2	P55, P94	P55-P94
	3	None	None
4	1	P5, P96, P333, P334, P335, P336, P337	P5-P96, P96-P333, P333-P334, P334-P335, P335-P336, P336-P337
	2	P333, P336, P337	P333-P336
	3	None	None
5	1	P35, P352	P35-P352
	2	None	None
	3	None	None
6	1	None	None
	2	P399, P400	P399-P400
	3	None	None

Table 4.4: Wayposts Identified and Shortcuts Generated

As it can be observed from table 4.4, not all clusters found from the groups of paths can produce wayposts. For example, in cluster 1 of group 1, where the only common document between all the paths of the cluster was the target document. In order to identify wayposts from a cluster there should be at least a minimum of 1 common document that appears in all path members of the cluster, excluding the target document. Since the above cluster did not satisfy this condition, no wayposts could be identified from it. Furthermore, as mentioned earlier, one limitation of our approach in identifying wayposts is that it cannot be applied to singleton clusters. Therefore, the singleton clusters found in the above groups of paths were unable to produce any wayposts. Despite these drawbacks, it can be seen from table 4.4, that the above approach can produce viable wayposts from clusters of similar paths as long as the following two conditions are satisfied:

1. A cluster must have a minimum of 2 paths in it.
2. There should be at the very least 1 common document that appears in all path members of the cluster, excluding the target document.

Furthermore, the shortcuts suggested from these wayposts clearly demonstrates its potential in shortening a user's navigational path. For example, the second path in cluster 1 of group 1 as shown in appendix E, requires a user to make 6 clicks in order to reach the target document P29. However, with the shortcuts suggested between documents P25, P27 and P29 a user could reach the same target document in 2 clicks and at the same time, any information with regards to a user's need which, may be contained in these wayposts are also preserved.

4.4 Evaluation

In this section, we describe a method to evaluate the shortcuts generated from the wayposts. Since, the scope of our work did not include the implementation of the suggested shortcuts to the School's website, it is difficult to gauge their true usefulness in improving users' navigation in the website. Therefore, here we present a qualitative evaluation of the suggested shortcuts. A useful shortcut is one that shortens the navigational path of a user to reach a specific target document. With respect to this, we present scenarios where the shortcuts generated using wayposts could be useful in shortening a user's navigational path. In order to do this, we sampled 3 different users' navigational paths where the suggested shortcuts in section 4.3 could be useful, from the groups of paths listed in appendix

E. Figure 4.5 below, illustrates each path as a case in which the suggested shortcut could be useful in shortening the navigational paths.

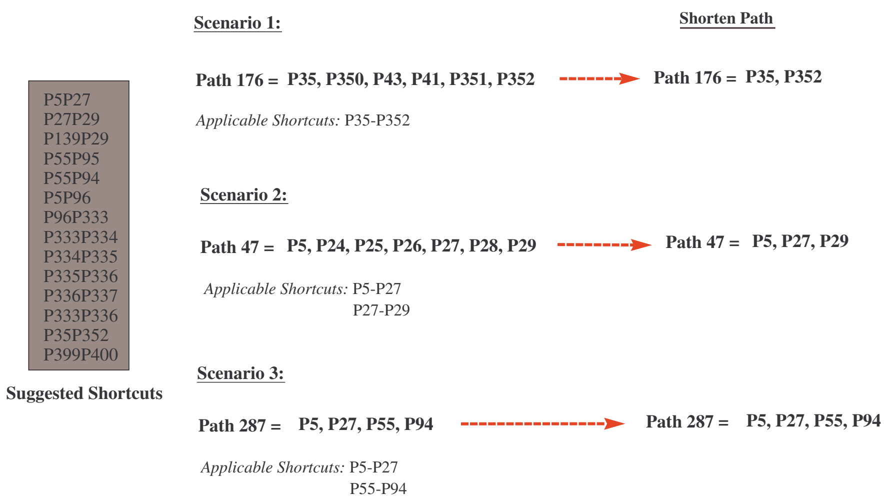

Scenario 1:
Shorten Path
Path 176 = P35, P350, P43, P41, P351, P352
Path 176 = P35, P352
Applicable Shortcuts: P35-P352
Scenario 2:
Path 47 = P5, P24, P25, P26, P27, P28, P29
Path 47 = P5, P27, P29
Applicable Shortcuts: P5-P27
P27-P29
Scenario 3:
Path 287 = P5, P27, P55, P94
Path 287 = P5, P27, P55, P94
Applicable Shortcuts: P5-P27
P55-P94
P5P27
P27P29
P139P29
P55P95
P55P94
P5P96
P96P333
P333P334
P334P335
P335P336
P336P337
P333P336
P35P352
P399P400
Suggested Shortcuts

Figure 4.5: How The Suggested Shortcuts May Be Useful in Shortening Users Navigational Paths

As shown in figure 4.5, there are two scenarios in which the shortcuts generated using wayposts could allow users to reach their target documents in shorter navigational paths, thereby reducing the time spend on browsing irrelevant document. Scenario 1, shows an example of a case where the target document of a website contains all the information required to satisfy a user's need. In such cases, it would be more useful to provide a link directly between the initial and target documents of the user's path bypassing the intermediate documents, similar to the existing studies. Note how the shortcuts suggested based on wayposts shown in figure 4.5 could cater for such a scenario by providing a link directly between the initial and target documents of the path 176 in scenario 1. Thus shortening the path required to reach the target document. On the other hand, scenario 2 shows an example of a case where the information required to satisfy a user's need may be spread across several documents rather than a single document. This could be due to a web designer deciding to break down a large source of information into several parts and distribute them across different documents to avoid creating one single long document. Longer documents may pose readability problems among some users, therefore it is a common practice for web designers to spilt the information into several shorter documents to make it easier for users to read. In figure 4.5, we can see how the shortcuts suggested using our approach could be useful in shortening the navigational paths corresponding to the above mentioned scenario. Finally, scenario 3 in figure 4.5 depicts a case where the shortcuts generated using our approach may not be so useful in shortening a user's navigational path. In this case, the users are already taking the most efficient path in order to reach a target document and therefore, the suggested shortcuts could not further reduce the length of the navigational path.

Existing studies tend to ignore the intermediate documents in a user's path and provide a shortcut between the initial and target documents of the user's path. This approach is useful in cases where the target document contains all the information required to satisfy a user's need. However, the effectiveness of the existing approach in a scenario where the information a user needs could be spread over several different documents in a website is still unproven. Our approach takes a different view on the intermediate documents appearing in a user's path, as we believe that they may contain information with regards to a user's need which could lead them to their target documents. Based on the above evaluation, it can be concluded that by taking these intermediate documents into consideration and identifying potential wayposts from them, we could suggest shortcuts which, not only shortens users' navigational paths to a target document, but also possibly reduce the loss of significant information that may be contained in the intermediate documents of a user's path. Hence, our approach could be used in both scenarios where the information a user needs can either be found in a single document or spread across several different documents.

4.5 Chapter Summary

In this chapter, we have described the process of discovering knowledge from the data acquired from the weblogs. The chapter highlights the method employed in identifying documents that are of interest to users and extracting a collection of navigational paths leading to those documents. Analysing these navigational paths reveals interesting information such as documents which are commonly visited among users' paths that lead to the same target document. We believed that these common documents could act as wayposts which, may be used to shorten users' navigational paths in a website. Therefore in this chapter, we presented means of identifying such wayposts and provided an evaluation of the shortcuts generated from them.

Chapter 5

Conclusions And Future Work

This project has investigated the idea of adaptive websites and presented an approach to adaptive websites using wayposts, which serve as navigational shortcuts that could be used to improve a website's organisation. In this chapter we conclude the thesis by highlighting our main contributions and identifying possible areas of future work.

5.1 Achievements of Objectives

Earlier in section 1.2 we had outlined 3 objectives of our research. In this section, we will examine our contributions with respect to these objectives.

1. *Identify a group of target documents based on users' access patterns and extract navigational paths terminating at those documents from users' sessions.* We have highlighted the challenges associated to finding a set of target documents from a website and indicated the importance of identifying such target documents correctly which, is critical for extracting navigational paths from a collection of users' sessions. In accordance to this, we presented the End Document method which identifies a set of potential target documents based on users' access patterns. The method uses the frequency with which a document appears as an end document in users' sessions and the number of its incoming visits to determine the candidacy of document as a target document. We have evaluated this method and shown that it is reasonably effective in finding target documents. To this extent, we have identified a set of

viable target documents using this method from the data acquired of the RGU School of Computing's web logs and extracted navigational paths leading to these target documents from users' sessions.

2. *Group similar navigational paths together and find hyperlinks (i.e., wayposts) that can act as potential shortcuts points by identifying documents that commonly co-occur in these paths.* One drawback of exisiting shorcutting methods is that they bypass the intermediate documents of a user's path and provide a shortcut between the initial and target document of the path. We have presented our approach which believes that these intermediate documents may contain information corresponding to a user's need that leads him to the target document. With regards to this, we have presented a way of identifying wayposts (i.e., intermediate documents) which can act as navigational shortcuts to target documents from a collection of users' paths. The method relies on grouping similar users' paths together and finding documents that co-occur in them. Documents which frequently co-occur in multiple users' paths which are similar indicate users interest in them, thereby making them a suitable candidate as wayposts. To this extent, we have implemented a k-means algorithm and used it to generate clusters of similar paths from the collection of users' paths found in the earlier objective. Then, for each cluster generated we determined the documents which co-occur in all path members of the cluster and identified them as wayposts which, can act as potential navigational shortcuts. We have shown that the clusters generated by the algorithm are reasonably good from which potential wayposts could be identified for shortcutting.

3. *Evaluating our approach by measuring the usefulness of the shortcuts generated from wayposts.* Since the objectives of our work does not include the incorporation of the suggested shortcuts to a website, it is not possible to gauge the actual usefulness of these shortcuts. Instead, we have presented a qualitative evaluation of some of the shortcuts generated and discussed scenarios in which they would be useful. Based on this, we have shown that the shortcuts generated using our approach could shorten users navigational paths to their target documents and reduce the possible loss of

information that may be contained within the intermediate documents of the users' paths.

5.2 Possible Extensions

The work carried out in this research has attempted to present an approach to adaptive websites with the aim of improving users' task of navigating in a website. However, this research is far from a finished work especially with regards to its long term goal of improving a website's organisation automatically. In this section, we will take a closer look at the limitations of the work reported here, identify ways of addressing these limitations and discuss possible extension to this research's scope.

Data acquisition from web logs. Data acquisition from web logs plays a vital role in usage mining as it influences the knowledge discovered from it. In this thesis, during the web logs filtration task we assumed that requests for documents other than HTML as irrelevant and filtered them out. This is despite the possibility that these non-html files could also be of a user's interest in the website. It is extremely difficult to determine if such non-html files are of importance or not based on the information available from web logs alone. The above mentioned assumption did not severely affect the outcome of this study, probably due to the likelihood that users could visit HTML documents containing links to such non-html files. However, one way to solve this issue could be to use additional website structure information to determine if these non-html files are part of a HTML document or a document by themself. Following that in the crawler logs elimination task, we described and employed a combined approach that uses a list of known crawlers IP addresses and the Robots Exclusion Standard to filter out crawler logs from web logs. Despite its simplicity, the experiment reported in this thesis proved the effectiveness the combined approach in detecting majority of the crawlers requests in the web logs. However, the Robots Exclusion Standard is not regulated by an accredited body and crawlers compliance to it is disputable. In addition, publicly available list of known crawlers IP addresses is not exhaustive. Therefore if a crawler is not listed and it does not comply with the Robots Exclusion Standard, it will be identified as a genuine user rather than a crawler. This could be addressed by employing a more sophisticated method of detecting a crawler's requests such as in (Tan and Kumar, 2002). Finally, in the sessionisation task, one major issue was the exclusion of the document requests belonging to the internal users made through the school's proxy server. These excluded requests constituted to approximately 65% of the original dataset and meant a possible bias in the outcome of this research. Yet, the research yielded fairly good results mainly because the remaining 35% of the dataset was made up of both internal and external users' document requests. However, this case may not hold true in other websites. Therefore one way to handle the document requests associated to proxy servers could be to the use of the proxy logs to distinguish document requests that were made by different users.

Limited background information. Our work relies solely on the knowledge discovered from the web logs. Although enriched with information on users' access patterns and website usage statistics, we have emphasized that the type of information that could be obtained from the CLF web logs are still limited. This introduced uncertainties in the approaches used to identify the end of a user's session and target documents from the website. Our work employs a time based approach where a threshold is used to identify both the end of a user's session and target documents from the web logs. However, this requires the information of the actual time users have spent in each document they had visited while browsing the website. The web logs used in our work does not capture such information. Therefore the time spent on a document by a user was approximated based on the date and time stamp of subsequent document requests appearing in the web logs. Although the approximation provides us with reasonable data to work with, it is far from being the actual time user's have spent on a document. This uncertainty also persist in the case of identifying the end of a user's session. The time thresholds set to indicate the end of a user's session was based on the TI distribution of users' logs and the same method mentioned above was used generate the distribution. This limitation could be addressed with the use of customised cookies written to assign a session ID for each visit made by a user to the website and capture the time spent on each document visited by him in the session.

As for the shortcuts proposed from wayposts, our work does not consider if a link already exists between two documents prior to suggesting a new shortcut between them. This may be addressed by exploiting the hyperlinks structure of the documents in the website, to provide additional information whether two documents are linked or not prior to suggesting a shortcut between them.

Dynamic Websites. The work reported in this thesis was based on a single website, which is the RGU's School of Computing website. The school's website mainly consisted of documents with static information in them and did not have any dynamic features. Therefore the applicability and performance of our approach on other websites such as dynamic websites still remains largely unknown.

Grouping similar paths together. In our work the measure employed to determine the similarity between two paths was focussed on the number of co-occurring documents between them. It did not take into account of the sequence in which these documents appeared in the paths. Therefore in future we could look into grouping the paths based on the sequence in which co-occurring documents appear between two paths. Furthermore our work relied on K-means algorithm to cluster similar paths together. As part of the future work, we could look into employing a conceptual based clustering technique such as COBWEB and compare the performances of these two algorithms in grouping similar paths together.

Qualitative Evaluation. We evaluated our approach using a qualitative method and discussed the scenario's in which the shortcuts could be useful in shortening users navigational paths. This method allowed us to give an account of how useful the shortcuts suggested could be in assisting the users to reach their target documents. However, in order to gauge the true usefulness of these shortcuts a quantitative method is required.

One possible suggestion for this would be the use of historical web logs to obtain an approximation of the usefulness of the suggested shortcuts. By dividing our main data set into training and test data, we could generate a set potential shortcuts from the training data. We then obtain a collections of users' paths from the test data based on the target documents identified in the training data. Following this, for each path discovered from the test data we could compute the number of original clicks a user requires to reach a specific target document. Then we can recompute the number of clicks a user could have saved from using the new shortcuts for each path discovered in the test data. This could allow us to obtain a quantitative approximation of the usefulness of the suggested shortcuts in shortening navigational paths taken by the users to reach their target documents.

Automated adaptation of website's organisation. In this thesis, we focussed on suggesting shortcuts based on the wayposts identified from frequently travelled users' paths. However, our long term goal is to further extend this work so that these proposed shortcuts could be automatically incorporated into the website. This involves in developing a tool that automates website transformation allowing a webmaster to monitor and manipulate the changes made to the website's organisation. In particular, we plan to look into means of adding the generated shortcuts to the website by addressing several key issues concerned with how, where and should the generated shortcuts be implemented to the website. A way forward with the first two issues would be to adopt the dedicated dialogue box adaptation method introduced in (Perkowitz and Etzioni, 1997). In this method, given a shortcut to be added between two documents A and B, a dedicated dialogue box with restricted space is placed in A to promote the hyperlink to B. New shortcuts are added into the dialogue box as they are suggested and unpopular shortcuts will be removed to make way for the new ones. As for the third issue of whether a shortcut should be added or not, it could be easily solved using the hyperlinks structure information of the website. Furthermore labelling the shortcuts is equally important to adding them to the website as improperly labelled shortcuts are disadvantageous. Therefore we might also look into the use of content mining techniques to generate meaningful labels for the shortcuts prior to adding them to the website.

5.3 Summary

Many have attempted to build Web technologies that could enable users to be more selective of the information provided to them. Conventional personalisation or recommender systems tend to only filter information on a website to suit the ever changing users' need. However, such technologies have not considered the true source of users' problems in the Web which, are website design related issues. In this project, we presented an approach to adaptive website through means of shortcutting to try to address website design related problems. Unlike conventional personalisation and recommender systems which performs superficial changes to a website, our approach identified wayposts based on users access patterns and suggested shortcuts from them which could be used to change a website's organisation. This continuous adaptation of a website's design based on the users behaviour in the website will provide them with an improved browsing experience compared to existing intelligent websites.

Bibliography

Agarwal, C. C. (2004). On leveraging user access patterns for topic specific crawling, *Data Mining and Knowledge Discovery* **9**(2): 123–145.

Altingovde, I. S. and Ulusoy, O. (2004). Exploiting interclass rules for focussed crawling, *IEEE Intelligent System* **19**(6): 66–73.

Ansari, S., Kohavi, R., Mason, L. and Zheng, Z. (2001). Integrating e-commerce and data mining: Architecture and challenges, *Proceedings of the 2001 IEEE International Conference on Data Mining*, IEEE Computer Society, pp. 27–34.

BBC (2005). Web enjoys year of biggest growth, Online. accessed on 05/2008. *http://news.bbc.co.uk/2/hi/technology/4325918.stm

Berendt, B., Mobasher, B., Nakagawa, M. and Spiliopoulou, M. (2002). The impact of site structure and user environment on session reconstruction in web usage analysis, *Proceedings of the 4th WebKDD 2002 Workshop, at the ACMSIGKDD Conference on Knowledge Discovery in Databases*, pp. 115–129.

Brickell, J., Dhillon, I. S. and Modha, D. S. (2007). Adaptive website design using caching algorithm, *Advances in Web Mining and Web Usage Analysis*, Vol. 4811/2007, Springer, pp. 1–20.

Catledge, L. D. and Pitkow, J. E. (1995). Characterizing browsing strategies in the World Wide Web, *Computer Networks and ISDN Systems* **27**(6): 1065–1073.

Chakrabarti, S., Dom, B. and van de Berg, M. (1999). Focussed crawling: A new approach to topic specific web resource discovery, *Proceedings of the 8th World Wide Web Conference*, Elsevier, Toronto, pp. 545–562.

Cooley, R., Mobasher, B. and Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns, *Knowledge and Information System* **1**(1): 5–32.

Cooley, R., Tan, P.-N. and Srivastava, J. (2000). Discovery of interesting usage patterns from web data, *Web Usage Analysis and User Profiling*, Vol. 1836/2000, Springer, San Diego, pp. 163–182.

Crescenzi, V., Merialdo, P. and Missier, P. (2005). Clustering Web pages based on their structures, *Data and Knowledge Engineering* **54**: 279–299.

Etzioni, O. and Perkowitz, M. (1997). Adaptive websites: An AI challenge, *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, pp. 16–23.

Etzioni, O. and Perkowitz, M. (2000). Towards adaptive web sites: Conceptual framework and case study, *Artificial Intelligence* **118**(1-2): 245–275.

Facca, F. and Lanzi, P. (2005). Mining interesting knowledge from weblogs: A survey, *Data Mining and Knowledge Discovery* **53**(5): 225–241.

Fan, L., Cao, P., Lin, W. and Jacobson, Q. (1999). Web prefetching between low-bandwidth clients and proxies: Potential and performance, *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ACM Press, New York, pp. 178–187.

Festa, P. (2003). Personalized websites are driving customers away, Online. accessed on 03/2007.
*http://networks.silicon.com/webwatch/0,39024667,10006394,00.htm

Godoy, D. and Amandi, A. (2006). Modeling user interests by conceptual clustering, *Information Systems*, Vol. 31, Elsevier Science Ltd., pp. 247–265.

Goker, A. and He, D. (2000a). Analysing Web search logs to determine session boundaries for user-oriented learning, *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, Vol. 1892/2000, Springer, pp. 319–322.

Goker, A. and He, D. (2000b). Detecting session boundaries from Web user logs, *Proceedings of the 22nd Annual Colloquium on Information Retrieval Research*, pp. 57–66.

Goker, A., He, D. and Harper, D. (2002). Combining evidence for automatic web session identification, *Information Processing and Management* **38**(5): 727–742.

Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations, *Proceedings of the 16th National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Menlo Park, CA, pp. 439–446.

Jiang, X.-M., Song, W.-G. and Zeng, H.-J. (2005). Applying associative relationship on the clickthrough data to improve web search, *Advances in Information Retrieval*, Vol. 3408/2005, Springer, pp. 475–486.

Langley, P. (1999). User modeling in adaptive interfaces, *Proceedings of the 7th International Conference on User Modeling*, Springer New York, Inc., pp. 357–370.

Lee, J.-H. and Shiu, W.-K. (2004). An adaptive website system to improve efficiency with web mining techniques, *Advanced Engineering Informatics* **18**(3): 129–142.

Lieberman, H. (1995). Letizia: An agent that assists Web browsing, *Proceedings of the International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, pp. 924–929.

Lieberman, H., Dyke, N. V. and Vivacqua, A. S. (1999). Let's Browse: A collaborative Web browsing agent, *Proceedings of the 1999 International Conference on Intelligent User Interfaces*, ACM Press, Los Angeles, CA, pp. 65–68.

Lourenco, A. and Belo, O. (2006). Catching web crawlers in the act, *Proceedings of the 6th International Conference on Web Engineering*, ACM, pp. 265–272.

Masseglia, F., Poncelet, P. and Teisseire, M. (2003). Using data mining techniques on Web access logs to dynamically improve hypertext structure, *SIGWEB Newsletter* **8**(3): 13–19.

McGovern, G. (2003). Why personalization hasn't worked, Online. accessed on 03/2007. *http://www.gerrymcgovern.com/nt/2003/nt_2003_10_20_personalization.htm

Mitchell, T., Joachims, T. and Freitag, D. (1997). Web watcher: A tour guide for the world wide web, *Proceeding of the 15th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Japan, pp. 770–775.

Mobasher, B., Cooley, R. and Srivastava, J. (1999). Creating adaptive web sites through usage-based clustering of URLs, *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*, IEEE Computer Society, Washington, USA, pp. 19–25.

Mobasher, B., Cooley, R. and Srivastava, J. (2000). Automatic personalization based on web usage mining, *Communications of ACM* **43**(8): 142–151.

Netcraft (2008). Web server survey, Online. accessed on 05/2008. *http://news.netcraft.com/archives/web_server_survey.html

Page, L., Brin, S., Motwani, R. and Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web, *Technical report*, Stanford Digital Library Technologies Project. citeseer.ist.psu.edu/page98pagerank.html.

Pavlov, D., Manvoglu, E., Giles, L. and Pennock, D. (2004). Collaborative filtering with maximum entropy, *IEEE Intelligent Systems* **19**(6): 40–48.

Perkowitz, M. and Etzioni, O. (1997). Adaptive sites: Automatically learning from user access patterns, *Technical report*, Department of Computer Science and Engineering, University of Washington.

Pirolli, P., Pitkow, J. and Rao, R. (1996). Silk from a sow's ear: Extracting usable structures from the Web, *Proceedings of the 1996 Conference on Human Factors in Computing Systems (CHI-96)*, Vancouver, British Columbia, Canada.

Pitkow, J. (1997). In search of reliable usage data on the WWW, *Computer Networks and ISDN Systems*, Vol. 29, Elsevier Science, Santa Clara, CA, pp. 1343–1355.

Rangarajan, S., Phoha, V., Balagani, K., Selmic, R. and Iyengar, S. (2004). Adaptive neural network clustering of Web users, *Computer* **37**(4): 34–40.

Shardanand, U. and Maes, P. (1995). Social information filtering:algorithms for automating "word of mouth", *Proceedings of ACM Conference on Human Factor in Computing Systems*, Vol. 1, ACM Press, pp. 210–217.

SPSS, I. (2008). Predictive web analytics, Online. accessed on 04/2008. *http://www.spss.com/pwa/index.html

Srikant, R. and Yang, Y. (2001). Mining web logs to improve website organization, *Proceedings of the 10th international conference on World Wide Web*, ACM Press, New York, NY, pp. 430–437.

Srivastava, J., Cooley, R., Deshpande, M. and Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data, *SIGKDD Explorations* **1**(2): 12–23.

Tan, P.-N. and Kumar, V. (2002). Discovery of web robot sessions based on their navigational patterns, *Data Mining and Knowledge Discovery* **6**: 9–35.

WebTrend, I. (2008). Webtrends visitor intelligence, Online. accessed on 04/2008. *http://www.webtrends.com/Products/WebTrendsVisitorIntelligence.aspx

Wexelblat, A. and Maes, P. (1999). Footprints: History-rich tools for information foraging, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, pp. 270–277.

Appendix A

Sample Web Logs

```
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/cow.html HTTP/1.1" 200 695
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/images/ribbons.gif HTTP/1.1" 200 32768
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/menu.html HTTP/1.1" 200 1140
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/menu.html HTTP/1.1" 200 1140
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/images/legback.gif HTTP/1.1" 200 8799
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/images/ribbons.gif HTTP/1.1" 200 0
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/cow_noframe.html HTTP/1.1" 200 2404
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/images/ribbons.gif HTTP/1.1" 200 0
villagegreen.windchannel.com - - [01/Oct/2004:00:00:56 +0100] "GET /staff/dc/hcow/images/legback.gif HTTP/1.1" 200 0
villagegreen.windchannel.com - - [01/Oct/2004:00:00:59 +0100] "GET /staff/dc/hcow/images/ribbons.gif HTTP/1.1" 206 65536
villagegreen.windchannel.com - - [01/Oct/2004:00:01:00 +0100] "GET /staff/dc/hcow/images/ribbons.gif HTTP/1.1" 200 32768
villagegreen.windchannel.com - - [01/Oct/2004:00:01:08 +0100] "GET /staff/dc/hcow/images/ribbons.gif HTTP/1.1" 206 32768
65.54.188.52 - - [01/Oct/2004:00:01:22 +0100] "GET /docs/jdk1.3/api/org/omg/CORBA/WCharSeqHelper.html HTTP/1.0" 200 13365
65.54.188.52 - - [01/Oct/2004:00:01:37 +0100] "GET /docs/jdk1.3/api/org/omg/CORBA/UnionMemberHelper.html HTTP/1.0" 200 13967
65.54.188.52 - - [01/Oct/2004:00:01:49 +0100] "GET /docs/jdk1.3/api/java/lang/reflect/Array.html HTTP/1.0" 200 54199
65.54.188.52 - - [01/Oct/2004:00:02:10 +0100] "GET /docs/jdk1.3/api/java/awt/RenderingHints.Key.html HTTP/1.0" 200 14301
65.54.188.52 - - [01/Oct/2004:00:02:17 +0100] "GET /support/support/gen-info/RGU_INFO_DISK.html HTTP/1.0" 404 1176
213-73-210-224.cable.quicknet.nl - - [01/Oct/2004:00:02:22 +0100] "GET /robots.txt HTTP/1.0" 200 346
65.54.188.52 - - [01/Oct/2004:00:02:23 +0100] "GET /support/support/Local/index.htm HTTP/1.0" 404 1176
65.54.188.52 - - [01/Oct/2004:00:02:30 +0100] "GET /support/support/FAQ/FAQtop.htm HTTP/1.0" 404 1176
65.54.188.52 - - [01/Oct/2004:00:02:35 +0100] "GET /support/support/Info/Infotop.htm HTTP/1.0" 404 1176
crawl-66-249-64-28.googlebot.com - - [01/Oct/2004:00:05:58 +0100] "GET /robots.txt HTTP/1.0" 200 346
crawl-66-249-64-28.googlebot.com - - [01/Oct/2004:00:05:58 +0100] "GET /research/group.php?id=CIG HTTP/1.0" 301 379
217.107.244.234 - - [01/Oct/2004:00:06:13 +0100] "GET /students/cs_yr94/lk/fido.html HTTP/1.1" 404 1188
```

```
217.107.244.234 - - [01/Oct/2004:00:06:13 +0100] "GET /logos/small_logo.gif HTTP/1.1" 404 1188
217.107.244.234 - - [01/Oct/2004:00:06:13 +0100] "GET /more_icons/small_logo2.gif HTTP/1.1" 200 2232
217.107.244.234 - - [01/Oct/2004:00:06:13 +0100] "GET /backgrounds/wave_rc.gif HTTP/1.1" 404 1188
bays264-0b01-dhcp076.bu.edu - - [01/Oct/2004:00:08:52 +0100] "GET /staff/jm HTTP/1.1" 301 372
bays264-0b01-dhcp076.bu.edu - - [01/Oct/2004:00:08:52 +0100] "GET /staff/jm/ HTTP/1.1" 200 1294
crawl-66-249-64-44.googlebot.com - - [02/Oct/2004:00:00:05 +0100] "GET /staff/jrc/SEAR.pdf HTTP/1.0" 200 440572
cr2.turnitin.com - - [02/Oct/2004:00:00:46 +0100] "GET /docs/jdk1.3/api/org/omg/CORBA/VM_ABSTRACT.html HTTP/1.0" 200 7590
cr2.turnitin.com - - [02/Oct/2004:00:02:00 +0100] "GET /docs/jdk1.3/api/org/omg/CORBA/VM_CUSTOM.html HTTP/1.0" 200 7993
tvg-ext-061-106.tvguideinc.com - - [02/Oct/2004:00:00:39 +0100] "GET /staff/dc/hcow/cow.html HTTP/1.1" 200 695
tvg-ext-061-106.tvguideinc.com - - [02/Oct/2004:00:00:39 +0100] "GET /staff/dc/hcow/menu.html HTTP/1.1" 200 1140
tvg-ext-061-106.tvguideinc.com - - [02/Oct/2004:00:00:39 +0100] "GET /staff/dc/hcow/cow_noframe.html HTTP/1.1" 200 2404
tvg-ext-061-106.tvguideinc.com - - [02/Oct/2004:00:00:39 +0100] "GET /staff/dc/hcow/images/legback.gif HTTP/1.1" 200 0
systemy74.toya.net.pl - - [02/Oct/2004:00:03:02 +0100] "GET /robots.txt HTTP/1.1" 200 346
systemy74.toya.net.pl - - [02/Oct/2004:00:03:02 +0100] "GET /research/kbs/kacbd/kacbd.html HTTP/1.1" 301 395
crawl-66-249-64-141.googlebot.com - - [02/Oct/2004:00:07:33 +0100] "GET /robots.txt HTTP/1.0" 200 346
crawl-66-249-64-141.googlebot.com - - [02/Oct/2004:00:07:34 +0100] "GET /staff/chb HTTP/1.0" 301 361
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:49 +0100] "GET / HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:49 +0100] "GET /images/spacer.gif HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:49 +0100] "GET /images/index_r1_c5.gif HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:50 +0100] "GET /images/start_r1_c1.gif HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:50 +0100] "GET /images/start_r1_c6.gif HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:50 +0100] "GET /images/start_r1_c4.gif HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:50 +0100] "GET /images/start_r2_c1.gif HTTP/1.1" 304 -
host217-137-242-12.no-dns-yet.ntli.net - - [02/Oct/2004:00:07:50 +0100] "GET /images/index_r3_c13.gif HTTP/1.1" 304 -
```

```
cache-loh-aa04.proxy.aol.com - - [02/Oct/2004:23:56:12 +0100] "GET /support/images/Supp_supphome.png HTTP/1.1" 200 954
cache-loh-aa04.proxy.aol.com - - [02/Oct/2004:23:56:12 +0100] "GET /support/images/supp_FAQ_email.png HTTP/1.1" 200 479
cache-loh-ab04.proxy.aol.com - - [02/Oct/2004:23:56:20 +0100] "GET /support/images/Supp_supphome_roll.png HTTP/1.1" 200 964
cache-loh-ab04.proxy.aol.com - - [02/Oct/2004:23:56:21 +0100] "GET /support/images/supp_FAQ_email_roll.png HTTP/1.1" 200 445
cache-loh-ab01.proxy.aol.com - - [02/Oct/2004:23:56:31 +0100] "GET /support/support.htm HTTP/1.1" 200 18035
cache-loh-aa08.proxy.aol.com - - [02/Oct/2004:23:56:32 +0100] "GET /support/images/supp_stats_roll.png HTTP/1.1" 200 1004
in7803b-198.staff.wlv.ac.uk - - [03/Oct/2004:00:02:12 +0100] "GET /tour/C26.htm HTTP/1.1" 200 12540
no-dns-yet.demon.co.uk - - [03/Oct/2004:00:12:28 +0100] "GET /favicon.ico HTTP/1.1" 404 1188
82.209.241.44 - - [03/Oct/2004:00:23:45 +0100] "GET /staff/sw/stuarts_papers/images/calman.gif HTTP/1.1" 200 3977
spider1.picsearch.com - - [03/Oct/2004:00:25:49 +0100] "GET /staff/dc/mmp/lecture3.html HTTP/1.0" 200 35632
proxy1ext.rgu.ac.uk - - [03/Oct/2004:00:26:40 +0100] "GET /support.htm HTTP/1.0" 200 18827
proxy1ext.rgu.ac.uk - - [03/Oct/2004:00:26:40 +0100] "GET /styles/openday.css HTTP/1.0" 200 1883
proxy1ext.rgu.ac.uk - - [03/Oct/2004:00:26:40 +0100] "GET /images/spacer.gif HTTP/1.0" 200 43
lj1069.inktomisearch.com - - [03/Oct/2004:00:27:41 +0100] "GET /robots.txt HTTP/1.0" 200 346
ex173.wc.ask.com - - [03/Oct/2004:00:29:37 +0100] "GET /staff/asga/mach.html HTTP/1.1" 200 12280
spider4.picsearch.com - - [03/Oct/2004:00:29:46 +0100] "GET /staff/dc/mmp/lecture1.html HTTP/1.0" 200 41093
ex173.wc.ask.com - - [03/Oct/2004:00:29:53 +0100] "GET /staff/dd/projects/rover/rover.html HTTP/1.1" 200 14679
spider3.picsearch.com - - [03/Oct/2004:00:31:53 +0100] "GET /staff/dc/mmp/lecture2.html HTTP/1.0" 200 20907
nig1.ng.britishcouncil.org - - [03/Oct/2004:00:32:51 +0100] "GET /images/start_r3_c1.gif HTTP/1.0" 304 -
nig1.ng.britishcouncil.org - - [03/Oct/2004:00:32:51 +0100] "GET /images/start_r3_c9.gif HTTP/1.0" 304 -
nig1.ng.britishcouncil.org - - [03/Oct/2004:00:32:51 +0100] "GET /images/index_r1_c5.gif HTTP/1.0" 304 -
```

Appendix B

Listings of Response Code

Content extracted from http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

“Informational 1xx

This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and is terminated by an empty line.

100: Continue

The client SHOULD continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server. The client SHOULD continue by sending the remainder of the request or, if the request has already been completed, ignore this response.

101: Switching Protocols

The server understands and is willing to comply with the client’s request, via the Upgrade message header field, for a change in the application protocol being used on this connection.

Successful 2xx

This class of status code indicates that the client's request was successfully received, understood, and accepted.

200: OK

The request has succeeded. The information returned with the response is dependent on the method used in the request, for example:

- GET an entity corresponding to the requested resource is sent in the response;
- HEAD the entity-header fields corresponding to the requested resource are sent in the response without any message-body;
- POST an entity describing or containing the result of the action;
- TRACE an entity containing the request message as received by the end server.

201: Created

The request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field.

202: Accepted

The request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place.

203: Non-Authoritative Information

The returned metainformation in the entity-header is not the definitive set as available from the origin server, but is gathered from a local or a third-party copy.

204: No Content

The server has fulfilled the request but does not need to return an entity-body, and might want to return updated metainformation.

205: Reset Content

The server has fulfilled the request and the user agent SHOULD reset the document view which caused the request to be sent.

206: Partial Content

The server has fulfilled the partial GET request for the resource. The request MUST have included a Range header field indicating the desired range, and MAY have included an If-Range header field to make the request conditional.

Redirection 3xx

This class of status code indicates that further action needs to be taken by the user agent in order to fulfill the request. The action required MAY be carried out by the user agent without interaction with the user if and only if the method used in the second request is GET or HEAD.

300: Multiple Choices

The requested resource corresponds to any one of a set of representations, each with its own specific location, and agent- driven negotiation information is being provided so that the user (or user agent) can select a preferred representation and redirect its request to that location.

301: Moved Permanently

The requested resource has been assigned a new permanent URI and any future references to this resource SHOULD use one of the returned URIs.

302: Found

The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client SHOULD continue to use the Request-URI for future requests.

303: See Other

The response to the request can be found under a different URI and SHOULD be retrieved using a GET method on that resource.

304: Not Modified

If the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server SHOULD respond with this status code.

305: Use Proxy

The requested resource MUST be accessed through the proxy given by the Location field. The Location field gives the URI of the proxy. The recipient is expected to repeat this single request via the proxy. 305 responses MUST only be generated by origin servers.

306: (Unused)

The 306 status code was used in a previous version of the specification, is no longer used, and the code is reserved.

307: Temporary Redirect

The requested resource resides temporarily under a different URI. Since the redirection MAY be altered on occasion, the client SHOULD continue to use the Request-URI for future requests.

Client Error 4xx

The 4xx class of status code is intended for cases in which the client seems to have erred. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method.

400: Bad Request

The request could not be understood by the server due to malformed syntax.
401: Unauthorized

The request requires user authentication.

402: Payment Required

This code is reserved for future use.

403: Forbidden

The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated.

404: Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address.

405: Method Not Allowed

The method specified in the Request-Line is not allowed for the resource identified by the Request-URI.

406: Not Acceptable

The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

407: Proxy Authentication Required

This code is similar to 401 (Unauthorized), but indicates that the client must first authenticate itself with the proxy.

408: Request Timeout

The client did not produce a request within the time that the server was prepared to wait.

409: Conflict

The request could not be completed due to a conflict with the current state of the resource.

410: Gone

The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent.

411: Length Required

The server refuses to accept the request without a defined Content- Length.

412: Precondition Failed

The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.

413: Request Entity Too Large

The server is refusing to process a request because the request entity is larger than the server is willing or able to process.

414: Request-URI Too Long

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

415: Unsupported Media Type

The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

416: Requested Range Not Satisfiable

A server SHOULD return a response with this status code if a request included a Range request-header field, and none of the range-specifier values in this field overlap the current extent of the selected resource, and the request did not include an If-Range request-header field.

417: Expectation Failed

The expectation given in an Expect request-header field could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.

Server Error 5xx

Response status codes beginning with the digit ”5” indicate cases in which the server is aware that it has erred or is incapable of performing the request.

500: Internal Server Error

The server encountered an unexpected condition which prevented it from fulfilling the request.

501: Not Implemented

The server does not support the functionality required to fulfill the request.

502: Bad Gateway

The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

503: Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

504: Gateway Timeout

The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request.

505: HTTP Version Not Supported

The server does not support, or refuses to support, the HTTP protocol version that was used in the request message.”

Appendix C

Target Documents Found Using Time Threshold

Page ID	Page Name
P1	/staff/dc/hcow/menu.html
P2	/staff/dc/hcow/cow_noframe.html
P5	/
P24	/welcome.htm
P25	/news/index.htm
P26	/about.htm
P27	/courses/index.htm
P29	/courses/Undergraduate/elective.htm
P42	/staff/smc/teaching/datamining/evaluation/
P46	/e-softtech/
P48	/e-softtech/calendar.htm
P49	/e-softtech/e_reqs.htm
P50	/e-softtech/fees.htm
P51	/docs/jdk1.3/api/java/lang/String.html
P56	/staff/rab/Databases/
P58	/staff/khui/
P71	/support.htm
P74	/contacts.htm
P75	/staff/nc/pages-of-nav-lecture-notes/CM2003.htm
P88	/staff/fh/CM1008/schedule.html
P89	/staff/fh/CM1008/Examples/WebCrawling.html
P99	/history.htm
P107	/courses/Undergraduate/cga.htm
P108	/school.htm
P109	/courses/Undergraduate/cim.htm
P110	/courses/Undergraduate/cma.htm
P111	/support/laptopsforstudents/index.htm
P113	/courses/Undergraduate/cbe.htm
P114	/courses/Undergraduate/cs.htm
P115	/e-softtech/qualifications.htm
P128	/courses/Postgraduate/mcie.htm
P129	/e-softtech/formpage.htm
P132	/courses/Postgraduate/mcst.htm
P824	/docs/jdk1.3/api/java/awt/geom/class-use/Point2D.Float.html
P825	/docs/jdk1.3/api/org/omg/CORBA/class-use/Policy.html
P826	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyError.html
P827	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyHelper.html
P828	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyHolder.html
P829	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyListHelper.html
P830	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyListHolder.html
P831	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyOperations.html
P832	/docs/jdk1.3/api/org/omg/CORBA/class-use/PolicyTypeHelper.html
P833	/docs/jdk1.3/api/javax/swing/event/class-use/PopupMenuEvent.html
P834	/docs/jdk1.3/api/javax/swing/event/class-use/PopupMenuListener.html
P835	/docs/jdk1.3/api/javax/swing/plaf/class-use/PopupMenuUI.html
P836	/docs/jdk1.3/api/javax/sound/sampled/class-use/Port.html
P837	/docs/jdk1.3/api/javax/sound/sampled/class-use/Port.Info.html
P838	/docs/jdk1.3/api/javax/rmi/CORBA/class-use/PortableRemoteObjectDelegate.html
P839	/docs/jdk1.3/api/javax/swing/text/class-use/Position.html
P840	/docs/jdk1.3/api/javax/swing/text/class-use/Position.Bias.html

Table C.1: A Small Sample of Target Documents Identified Based on TimeThreshold

Appendix D

Examples of Paths Extracted Based on Identified Target Documents

Path ID	Path Sequence
1	P5, P24, P25, P26, P27, P28, P29,
2	P0, P1, P2, P2, P52,
3	P2, P69, P1, P0, P52,
4	P72, P72, P73,
5	P5, P24, P90, P91,
6	P5, P27, P91,
7	P5, P5, P27, P24, P133, P87,
8	P87, P28, P91,
9	P91, P61, P90, P27, P27, P24, P134, P5, P135, P136,
10	P91, P133, P145, P134, P138, P146,
11	P29, P25, P147, P26, P137, P139, P26, P148, P87,
12	P87, P28, P64, P135, P149, P150, P88, P62, P90, P151, P136,
13	P136, P25, P152, P95,
14	P95, P138, P90, P91,
15	P91, P137, P153, P154, P155, P94,
16	P94, P154, P162, P134, P29,
17	P146, P163, P164,
18	P164, P165, P26, P148, P28, P149, P166, P167, P150, P62, P168, P151, P169, P136,
19	P136, P25, P152, P170, P90, P91,
20	P91, P137, P155, P94,
21	P181, P181, P182,
22	P5, P5, P27, P24, P91,
23	P5, P24, P87,
24	P27, P24, P87,
25	P56, P155, P166, P172, P146,
26	P5, P24, P87,
27	P5, P24, P87,
28	P5, P24, P91,
29	P24, P232, P251, P322,
30	P181, P324, P325,
31	P5, P24, P26, P5, P91,
32	P5, P27, P87,
33	P5, P332, P290, P96, P333, P334, P335, P336, P337,
34	P96, P290, P332, P340,
35	P79, P80, P81, P179,
36	P129, P72, P73,
37	P129, P93, P344,
38	P129, P79, P80, P81, P179,
39	P129, P79, P81, P80, P179,
40	P129, P79, P80, P81, P178,
41	P129, P79, P81, P80, P178,
42	P129, P72, P73,
43	P129, P79, P81, P80, P180, P178,
44	P129, P79, P81, P80, P179,
45	P347, P319, P348, P349, P325,
46	P5, P35, P350, P43, P351, P41, P352,
47	P352, P40, P40, P35, P350, P43, P352,
48	P5, P96, P333, P334, P335, P336, P337,
49	P87, P56, P164,
50	P5, P24, P87,
51	P87, P232, P245, P244, P243, P253, P353, P354, P355, P356,
52	P27, P139, P29,
53	P357, P339, P340,

Table D.1: Small Sample of Path Extracted from Identified Target documents

Appendix E

Clusters of Similar Paths

Group 1: 8 Paths leading to Target Doc: P29

Cluster No: 1

Path ID: 76| P5, P27, P139, P29,

Path ID: 0| P5, P24, P25, P26, P27, P28, P29,

Path ID: 17| P94, P154, P162, P134, P29,

Cluster No: 2

Path ID: 382| P5, P27, P134, P29,

Path ID: 197| P5, P27, P134, P29,

Path ID: 282| P5, P27, P139, P29,

Cluster No: 3

Path ID: 53| P27, P139, P29,

Path ID: 58| P291, P139, P29,

Group 2: 13 Paths leading to Target Doc: P95

Cluster No: 1

Path ID: 56| P5, P27, P55, P95,

Path ID: 139| P5, P5, P27, P55, P95,

Path ID: 160| P5, P27, P55, P95,

Path ID: 171| P5, P27, P55, P95,

Path ID: 249| P5, P27, P55, P95,

Path ID: 267| P87, P147, P155, P27, P139, P134, P55, P95,

Path ID: 289| P5, P24, P27, P55, P95,

Path ID: 315| P5, P27, P55, P95,

Path ID: 386| P5, P27, P61, P63, P55, P95,

Path ID: 394| P5, P27, P55, P95,

Path ID: 429| P5, P27, P55, P95,

Cluster No: 2

Path ID: 173| P139, P55, P95

Cluster No: 3

Path ID: 13| P136, P25, P152, P95,

Group 3: 29 Paths leading to Target Doc: P94

Cluster No: 1

Path ID: 21| P91, P137, P155, P94,

Path ID: 15| P91, P137, P153, P154, P155, P94,

Path ID: 16| P94, P156, P24, P157, P158, P135, P28, P159, P56, P63, P141, P143, P5, P30, P55, P160, P24, P161, P134, P65, P94,

Cluster No: 2

Path ID: 287| P5, P27, P55, P94,

Path ID: 108| P5, P27, P55, P94,

Path ID: 117| P5, P27, P55, P94,

Path ID: 132| P5, P27, P55, P94,

Path ID: 134| P5, P27, P55, P94,

Path ID: 136| P5, P27, P55, P94,

Path ID: 143| P5, P27, P55, P94,

Path ID: 149| P5, P5, P27, P55, P94,

Path ID: 187| P5, P27, P55, P94,

Path ID: 189| P5, P27, P55, P94,

Path ID: 190| P27, P55, P94,

Path ID: 230| P5, P5, P27, P55, P94,

Path ID: 231| P27, P55, P94,

Path ID: 232| P5, P27, P55, P94,

Path ID: 255| P5, P27, P55, P94,

Path ID: 296| P27, P55, P94,

Path ID: 322| P5, P27, P5, P55, P24, P94,

Path ID: 335| P5, P5, P27, P55, P94,

Path ID: 349| P5, P27, P55, P94,

Path ID: 362| P5, P27, P55, P94,

Path ID: 363| P5, P27, P55, P94,

Path ID: 373| P5, P27, P55, P94,

Path ID: 417| P61, P159, P55, P94,

Path ID: 422| P5, P27, P55, P94,

Path ID: 444| P5, P55, P94,

Cluster No: 3

Path ID: 158| P5, P27, P55, P94,

Group 4: 45 Paths leading to Target Doc: P337

Cluster No: 1

Path ID: 34| P5, P332, P290, P96, P333, P334, P335, P336, P337,

Path ID: 49| P5, P96, P333, P334, P335, P336, P337,

Cluster No: 2

Path ID: 82| P333, P334, P335, P336, P337,

Path ID: 55| P333, P335, P334, P336, P337,

Path ID: 57| P87, P24, P333, P334, P335, P336, P337,

Path ID: 60| P333, P334, P335, P359, P336, P337,

Path ID: 67| P333, P334, P335, P336, P337,

Path ID: 68| P333, P334, P335, P336, P337,

Path ID: 69| P333, P335, P334, P336, P337,

Path ID: 71| P333, P334, P335, P336, P337,

Path ID: 72| P333, P335, P336, P337,

Path ID: 78| P333, P335, P334, P336, P337,

Path ID: 80| P5, P24, P333, P334, P335, P336, P359, P337,

Path ID: 81| P337, P333, P336, P337,

Path ID: 119| P333, P334, P335, P336, P337,

Path ID: 121| P333, P334, P335, P336, P337,

Path ID: 209| P5, P508, P333, P334, P335, P336, P337,

Path ID: 236| P5, P333, P334, P335, P335, P336, P335, P337,

Path ID: 240| P333, P334, P335, P336, P337,

Path ID: 241| P333, P334, P335, P336, P337,

Path ID: 244| P333, P334, P335, P336, P337,

Path ID: 251| P5, P333, P334, P335, P336, P337,

Path ID: 256| P333, P334, P335, P359, P336, P337,

Path ID: 260| P333, P334, P335, P336, P337,

Path ID: 263| P333, P334, P335, P336, P337, Path ID: 268| P333, P334, P335, P359, P336, P337,

Path ID: 278| P333, P334, P335, P336, P337,

Path ID: 284| P333, P334, P335, P336, P337,

Path ID: 298| P333, P334, P335, P336, P337,

Path ID: 299| P5, P333, P334, P335, P336, P337, Path ID: 326| P333, P334, P335, P336, P337,

Path ID: 330| P340, P339, P639, P339, P367, P339, P357, P333, P334, P335, P335, P359, P336, P335, P337,

Path ID: 352| P333, P334, P335, P336, P337,

Path ID: 368| P333, P334, P335, P336, P337,

Path ID: 374| P333, P334, P335, P336, P337,

Path ID: 383| P333, P334, P335, P336, P337,

Path ID: 389| P5, P333, P334, P335, P336, P337,

Path ID: 401| P5, P24, P333, P334, P335, P336, P337,

Path ID: 403| P333, P334, P335, P336, P337,

Path ID: 411| P333, P334, P335, P336, P337,

Path ID: 414| P333, P334, P335, P335, P336, P335, P337,

Path ID: 416| P333, P334, P335, P336, P337,

Path ID: 439| P5, P387, P24, P379, P2105, P35, P41, P614, P2460, P2461, P2459, P2462, P2463, P2458, P2490, P333, P334, P335, P359, P336, P98, P337,

Path ID: 442| P5, P333, P334, P335, P336, P337,

Cluster No: 3

Path ID: 333| P745, P98, P337,

Group 5: 10 Paths leading to Target Doc: P352

Cluster No: 1

Path ID: 70| P5, P35, P350, P352,

Path ID: 47| P5, P35, P350, P43, P351, P41, P352,

Path ID: 120| P35, P41, P352,

Path ID: 176| P35, P350, P43, P41, P351, P352,

Path ID: 222| P156, P159, P65, P35, P350, P377, P41, P352,

Path ID: 293| P35, P351, P5, P27, P5, P40, P352,

Path ID: 385| P5, P35, P352,

Path ID: 438| P35, P350, P352,

Cluster No: 2

Path ID: 48| P352, P40, P40, P35, P350, P43, P352,

Cluster No: 3

Path ID: 440| P35, P43, P44, P125, P352,

Group 6: 11 Paths leading to Target Doc: P400

Cluster No: 1

Path ID: 276| P501, P502, P400,

Cluster No: 2

Path ID: 87| P5, P399, P400,

Path ID: 102| P87, P147, P399, P400,

Path ID: 104| P372, P374, P373, P476, P477, P478, P479, P399, P400,

Cluster No: 3

Path ID: 92| P399, P332, P400,

Path ID: 184| P332, P399, P400,

Path ID: 192| P399, P502, P400,

Path ID: 246| P498, P500, P400,

Path ID: 270| P5, P399, P400,

Path ID: 281| P399, P502, P400,

Path ID: 397| P87, P147, P399, P400,