CLUSTERING FREQUENT NAVIGATION PATTERNS FROM WEBSITE LOGS USING ONTOLOGY AND TEMPORAL INFORMATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEFA KILIÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

DECEMBER 2011

Approval of the thesis:

CLUSTERING FREQUENT NAVIGATION PATTERNS FROM WEBSITE LOGS USING ONTOLOGY AND TEMPORAL INFORMATION

submitted by **SEFA KILIÇ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** ________________

Assoc. Prof. Pınar Şenkul
Supervisor, **Computer Engineering Dept., METU** ________________

Prof. Dr. İsmail Hakkı Toroslu
Co-supervisor, **Computer Engineering Dept., METU** ________________

Examining Committee Members:

Assoc. Prof. Dr. Tolga Can
Computer Engineering Dept., METU ________________

Assoc. Prof. Dr. Pınar Şenkul
Computer Engineering Dept., METU ________________

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering Dept., METU ________________

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Dept., METU ________________

Güven Fidan, M.Sc.
AGMLab Information Technologies ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: SEFA KILIÇ

Signature :

ABSTRACT

CLUSTERING FREQUENT NAVIGATION PATTERNS FROM WEBSITE LOGS USING ONTOLOGY AND TEMPORAL INFORMATION

Kılıç, Sefa
M.S., Department of Computer Engineering
Supervisor : Assoc. Prof. Pınar Şenkul
Co-Supervisor : Prof. Dr. İsmail Hakkı Toroslu

December 2011, 73 pages

Given set of web pages labeled with ontological items, the level of similarity between two web pages is measured using the level of similarity between ontological items of pages labeled with. Using similarity measure between two pages, degree of similarity between two sequences of web page visits can be calculated as well. Using clustering algorithms, similar frequent sequences are grouped and representative sequences are selected from these groups. A new sequence is compared with all clusters and it is assigned to most similar one. Representatives of the most similar cluster can be used in several real world cases. They can be used for predicting and prefetching the next page user will visit or for helping the navigation of user in the website. They can also be used to improve the structure of website for easier navigation. In this study the effect of time spent on each web page during the session is analyzed.

Keywords: data mining, web usage mining, semantic similarity, clustering, web page recommendation

ÖZ

VARLIKBİLİM VE SÜRE BİLGİSİNİ KULLANARAK WEB SAYFALARINDAN SIK GÖRÜLEN DESEN KÜMELERİNİN ELDE EDİLMESİ

Kılıç, Sefa
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Pınar Şenkul
Ortak Tez Yöneticisi : Prof. Dr. İsmail Hakkı Toroslu

Aralık 2011, 73 sayfa

Varlikbilim öğeleri ile etiketlenmiş web sayfalarından oluşan bir kümede, iki web sayfası arasındaki benzerliğin derecesi, o sayfaları etiketlemekte kullanılan varlıkbilim öğelerinin arasındaki benzerlik kullanarak belirlenir. İki sayfa arasındaki benzerlik kullanılarak, gezilen web sayfalarından oluşan iki dizi arasındaki benzerlik de bulunabilir. Kümeleme algoritmalarını ile sık görülen benzer diziler gruplanır ve her kümeyi temsil edecek diziler seçilir. Yeni bir dizi geldiğinde, en benzer olan kümeye atanır. En benzer kümeyi temsil eden diziler bir çok gerçek senaryoda kullanılabilinir. Kullanıcının bir sonra ziyaret edeceği web sayfasının tahmin edilmesi ve önceden getirilmesi veya kullanıcıya daha kolay gezinme için yardım edilmesi, web sitesinin daha kolay gezinme için yapısının değiştirilmesi için kullanılabilir. Bu çalışmada web sayfalarında geçirilen sürenin oturumları kümeleme üzerindeki etkisi incelendi.

Anahtar Kelimeler: veri madenciliği, web kullanım madenciliği, anlamsal benzerlik, kümeleme, web sayfası önerme

To Doğa

ACKNOWLEDGMENTS

I would like to thank my advisors Dr. Pınar Şenkul and Dr. İsmail Hakkı Toroslu for their guidance through this study. I also would like to thank Sinem Demirci and Şükrü Bezen for their help during my study.

I also want to thank my family for their unconditional love and support.

I would like to thank friends from METU Department of Computer Engineering, especially to Çelebi Kocair, Onur Deniz, Erdal Sivri, Can Eroğul, Hande Çelikkanat, Selma Süloğlu, Anıl Sınacı for their friendship and great help during my study.

This study was supported by TÜBİTAK via the Support Programme for Scientific and Technological Research Projects (1001).

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

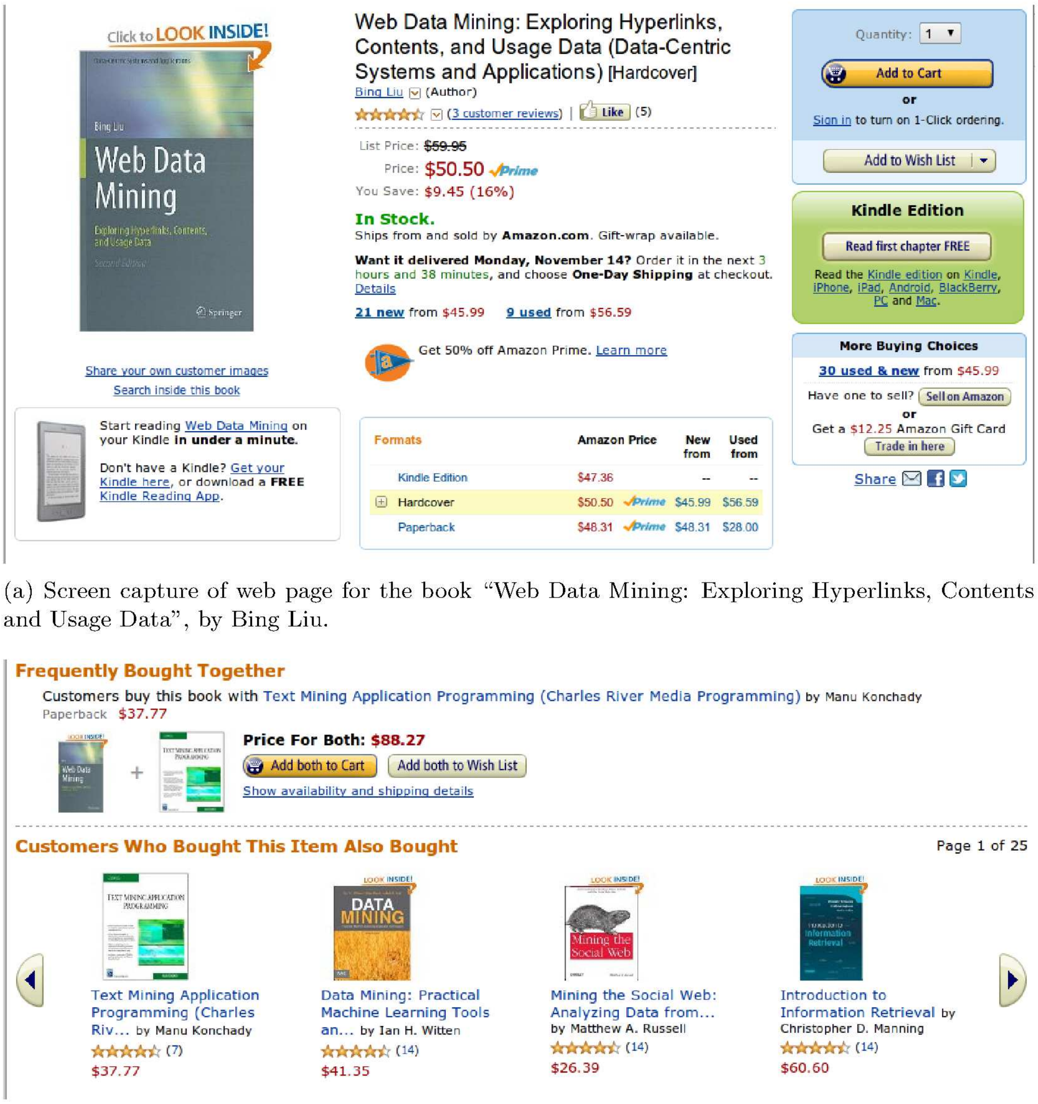
INTRODUCTION

Every year, the amount of documents on the Internet is increasing, as well as Internet users. Instead of bothering users with huge amount of data on a site, presenting the correct information at the right time in the most appropriate form is important and it results with better browsing experience for users [1].

Web mining methods are used for several different purposes. One important application area is web personalization which is the task of making web pages dynamically customized, based on characteristics of individual users. It is especially used by e-commerce applications to understand interests of users and to recommend products and show advertisements on the web page based on their preferences [2, 3]. Figure 1.1 shows a web page from e-commerce company Amazon.com. Product web pages are dynamically customized based on associations with other products and browsing history of the user.

Another recommendation example is given in Figure 1.2 from online movie database IMDb (Internet Movie Database). In this web site, when the user browses the page of a movie, movie recommendations are dynamically generated. Recommendations are generated based on genre, cast, writer, director information of the movie browsed and navigation history of the user on the whole site.

Another application area of web mining is the dynamic web page recommendation. Based on user navigation, next web pages that are likely to be requested by user are recommended to him/her. For example, in a web site, if users usually access page B from page A, B can be recommended to a user who is on page A. A similar important



Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications) [Hardcover]

Bing Liu (Author)

(3 customer reviews) | Like (5)

List Price: ~~$59.95~~

Price: $50.50 Prime

You Save: $9.45 (16%)

In Stock.

Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Monday, November 14? Order it in the next 3 hours and 38 minutes, and choose **One-Day Shipping** at checkout. Details

21 new from $45.99 **9 used** from $56.59

Get 50% off Amazon Prime. Learn more

or

Sign in to turn on 1-Click ordering.

Kindle Edition

Read the Kindle edition on Kindle, iPhone, iPad, Android, BlackBerry, PC and Mac.

More Buying Choices

30 used & new from $45.99

Have one to sell?

or

Get a $12.25 Amazon Gift Card

Share your own customer images

Search inside this book

Start reading Web Data Mining on your Kindle **in under a minute.**

Don't have a Kindle? Get your Kindle here, or download a **FREE** Kindle Reading App.

Formats	Amazon Price		New from	Used from
Kindle Edition	$47.36		--	--
⊞ Hardcover	$50.50	Prime	$45.99	$56.59
Paperback	$48.31	Prime	$48.31	$28.00

(a) Screen capture of web page for the book “Web Data Mining: Exploring Hyperlinks, Contents and Usage Data”, by Bing Liu.

Frequently Bought Together

Customers buy this book with Text Mining Application Programming (Charles River Media Programming) by Manu Konchady
Paperback $37.77

Price For Both: $88.27

Show availability and shipping details

Customers Who Bought This Item Also Bought

Page 1 of 25

Text Mining Application Programming (Charles Riv... by Manu Konchady
(7)
$37.77

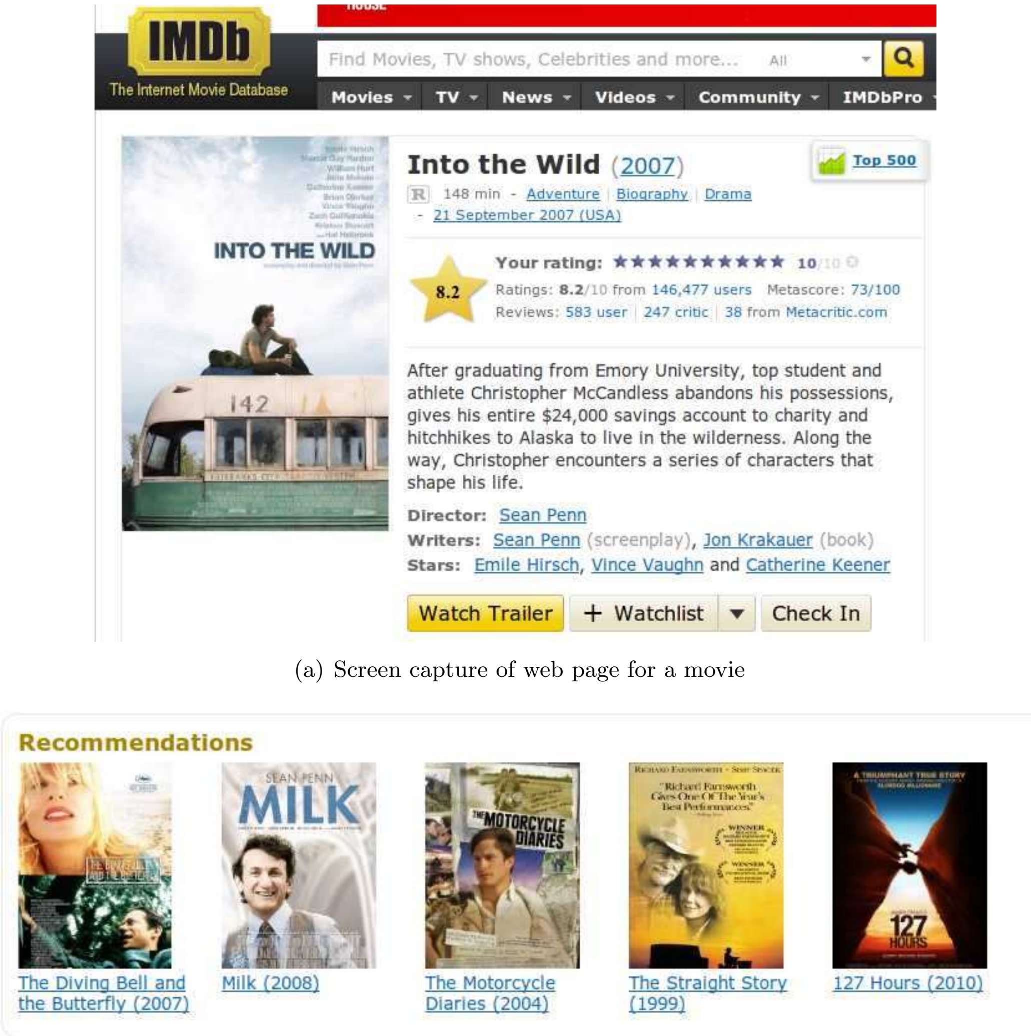
Data Mining: Practical Machine Learning Tools an... by Ian H. Witten
(14)
$41.35

Mining the Social Web: Analyzing Data from... by Matthew A. Russell
(14)
$26.39

Introduction to Information Retrieval by Christopher D. Manning
(14)
$60.60

(b) Recommendations for that book on the same page.

Figure 1.1: Sample recommendation from `www.amazon.com`



IMDb

The Internet Movie Database

Find Movies, TV shows, Celebrities and more... All

Movies TV News Videos Community IMDbPro

Into the Wild (2007)

Top 500

R 148 min - Adventure | Biography | Drama - 21 September 2007 (USA)

Your rating: ★★★★★★★★★★ 10/10

8.2

Ratings: 8.2/10 from 146,477 users Metascore: 73/100

Reviews: 583 user | 247 critic | 38 from Metacritic.com

After graduating from Emory University, top student and athlete Christopher McCandless abandons his possessions, gives his entire $24,000 savings account to charity and hitchhikes to Alaska to live in the wilderness. Along the way, Christopher encounters a series of characters that shape his life.

Director: Sean Penn

Writers: Sean Penn (screenplay), Jon Krakauer (book)

Stars: Emile Hirsch, Vince Vaughn and Catherine Keener

Watch Trailer | + Watchlist | Check In

(a) Screen capture of web page for a movie

Recommendations

The Diving Bell and the Butterfly (2007)

Milk (2008)

The Motorcycle Diaries (2004)

The Straight Story (1999)

127 Hours (2010)

(b) Recommendations of similar moviews on the same page

Figure 1.2: Sample recommendation from `www.imdb.com`

usage is prefetching and caching [3]. In the previous example, when the user browses page A, next page is likely to be B. While user is browsing A, B is fetched and cached. If user requests B, the copy in the cache is directly displayed without waiting the server for content. The aim is to make browsing faster for the user.

Hyperlinks can dynamically be inserted into web pages. Consider the following example. If there is a significant web navigation pattern of pages $A \to B \to C \to D$ and time spent on B and C are considerably small, it can be interpreted that users follow this navigation path to access page D from A. In this case, inserting a link from A to D may make the browsing easier for users. As well as dynamic hyperlink generation, the navigation patterns can also be used for evaluation of quality of the website [3]. Design and structure of a web site is important and it must be efficient, especially in e-commerce domain. A bad design of an e-commerce website may result with loss of potential customers, because of the structure that is not easy to explore by users. In addition to that, confused user navigations blur the statistics about pages which are also important for other analysis [4].

In this study, collected web navigation paths are clustered based on semantic similarity between paths. Resulting clusters can be used further for different purposes such as recommendation, prefetching of web pages or evaluating the overall quality of website structure. Contributions of this study are

- combining concept-based sequence clustering with time-spent information,
- different concept similarity metric,
- web usage mining on a non-comercial web domain, previous studies are usually on commercial web sites [5].

The rest of the thesis is organized as follows.

- Chapter 2 gives introduction to concepts web mining, steps of web mining process and pattern discovery methods.
- Chapter 3 describes the method used in this study.

- Chapter 4 is about methods used for experiments, results and discussion on them.
- Chapter 5 is about conclusions and future work on this study.

CHAPTER 2

BACKGROUND

Web mining is the analysis and extraction of meaningful and useful patterns from data on the World Wide Web [6].

Web mining is an active field of data mining for more than a decade. Cooley et al. give definition of web mining, techniques and issues on the topic [6]. They also give general architecture of WEBMINER, one of the earliest web mining system presented in [7]. In [8], Srivastava et al. give the taxonomy of web mining area and they discuss preprocessing, pattern discovery and analysis issues. They also give a survey about early research projects and commercial applications about web mining and they give an overview of WebSIFT as a web usage mining system [9]. One another contribution of their paper is the discussion on privacy issues on collection and use of user data by site administrators. The ethical issue is also discussed in [10]. Some reviews of the field are [1, 2, 11, 3]. Also, the book chapter by Mobasher is an extensive summary on web usage mining [12]. It covers all key steps of mining process including data collection, preprocessing, data modeling, discovery and analysis of usage patterns.

In [8], web mining is divided into three subfields which are content, structure and usage mining. These subfields are explained in following subsections.

2.1 Web Content Mining

Web content mining is the process to discover important information from the content in web pages. In past years, the content is usually text. However the amount of multimedia such as image, video and audio in the web increased so much in recent years.

For processing of text in web pages, the methods from natural language processing (NLP) and information retrieval (IR) is adopted. However, since the web is so huge and highly dynamic, these methods should be modified to meet requirements. Multimedia data should also be handled. To process multimedia data such as image, video and audio, it can be benefited from fields such as image processing, speech recognition etc.

2.2 Web Structure Mining

Web structure mining is the extraction of meaningful information using the structure of web pages. Two different types of structure are used to capture information. The first one is the hyperlink structure between web pages (i.e. inter-page structure [8]). Link-based classification, link-based cluster analysis are some possible tasks of web structure mining [13]. HITS [14] and PageRank [15] are some methods used in web structure mining to find the importance of web pages using link information.

The second type of structure is the one within the document (i.e. intra-page structure [8]). The pages are composed with markup languages such as HTML or XML in machine-readable form. Using intra-page structure of a page, information can be extracted from the page. For example, the HTML heading tags `<h1>` to `<h6>` are used to define headings. They are usually used for start a new section. Headings give important clues about text. Therefore heading information can be used and it is extracted from intra-structure of the web page. Another example is the HTML bold tag `<b>`. It is used to render text as bold. In a page, text between `<b>` and `</b>` can be considered as important in the page.

2.3 Web Usage Mining

In Web usage mining, the web page visit information of users is used. While the web content and structure mining use the real data on the web, web usage mining methods use data produced from behaviors of users [11]. For web usage mining, user-specific data are also used and it can be combined with content and/or structure data.

2.4 Data Types

All three subfields of web mining (i.e. web content/structure/usage mining) use web data to extract and analyze useful information, but data type used is different for each of them. Types of data used in web mining are content, structure and usage and user data [8].

- Content data in a site is usually text and multimedia data in the form of HTML/XML pages.
- Structure data is about the organization of pages in the web site. The structure data is usually captured from hyperlinks between web pages.
- Usage data contains not only web data but also the usage of it. The most common form of usage data is server logs which contain each request to each web page from different users. Log data usually contains time and date of the request, the IP address from which request comes from and request status etc. Usage data items that can be used for mining are further explained in Section 2.5.
- User data can also be helpful for mining process. Data such as demographic information and other profile information can be collected with a registration mechanism [12]. Also, user ratings, comments, purchase history and other user-specific information may be available and helpful [12].

2.5 Data Sources

The data collection is an important stage in the whole process, especially for web usage mining. The data to be used can be collected in server side or client side. Sometimes some intermediary sources are used for data collection too.

2.5.1 Server-side Log Collection

Server-side collection of data is the most common one among web mining studies in the literature. To collect data, server access logs, cookies and web bugs are commonly

Table 2.1: Example server access log

<ip> <userid> <request time> <request method/resource/protocol> <status code> <size> <referer> <agent>
1.2.3.4 - [06/May/2011:11:19:15 +0200] ‘‘GET A.html HTTP/1.1’’ 200 135 - ‘‘Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 3.5.20706)’’
1.2.3.4 - [06/May/2011:11:22:53 +0200] ‘‘GET B.html HTTP/1.1 200 257 A.html ‘‘Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 3.5.20706)’’
1.2.3.4 - [06/May/2011:11:23:01 +0200] ‘‘GET C.html HTTP/1.1’’ 200 232 B.html ‘‘Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 3.5.20706)’’
1.2.3.5 - [07/May/2011:19:58:30 +0200] ‘‘GET D.html HTTP/1.1’’ 200 103 - ‘‘Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.8) Gecko/20100214 Ubuntu/9.10 (karmic) Firefox/3.5.8’’
1.2.3.5 - [07/May/2011:19:58:48 +0200] ‘‘GET C.html HTTP/1.1’’ 200 232 D.html ‘‘Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.8) Gecko/20100214 Ubuntu/9.10 (karmic) Firefox/3.5.8’’
1.2.3.6 - [08/May/2011:08:23:12 +0200] ‘‘GET C.html HTTP/1.1’’ 200 232 www.example.com ‘‘Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.6) Gecko/20100205 Gentoo Firefox/3.5.6’’

used.

Server Access Logs One of the most popular ways to collect data is to use server access logs and in this study server access logs are used. Server access log file stores all requests to web pages of a site, usually in a standardized text file format such as Common or Extended log formats. One of the most commonly used data formats is Apache HTTP server combined log format. Portion of an example server access log file is given in Table 2.1. For each log entry, the first field is the IP address of the client requested to the server. The second field is the user id of the person requested resource. This part is “-” in all of example log entries, because none of resources are password protected. The next field is the time that the request was received. The fourth field contains the method used by the client, the address of resource requested and the protocol used. Next two fields are status code that server sends back to client and the size of object returned to the client. Last two fields store address of referer and information about the client browser. The referer address is the site that the client referred from, this referer site should have link to resource in third field.

Although web server logs are commonly used data sources, there are some disadvantages of using server logs [1]. The first one is web caching which is used to reduce traffic, latency and server load. Previously requested web documents such as HTML files, images are stored for a certain period of time. During this time, if temporarily stored files are requested again, stored ones are loaded, instead of new requests from web server. The caching mechanism can be implemented in client or proxy server level. Consider the web navigation sequence of a user, $A \rightarrow B \rightarrow C \rightarrow B \rightarrow D$. If web caching is used, the second request for page B is loaded from local storage, not from web server. Therefore, in server logs, the navigation sequence looks like $A \rightarrow B \rightarrow C \rightarrow D$. Resulting sequence negatively effects further steps of mining process. There may be no direct link from page C to D, or even if there exists, it may cause misinterpretations about usage data.

The second issue with web server logs is IP addresses. Due to usage of proxy servers requests from different machines are logged with the same IP address by the server. On the other hand, a user may be logged with multiple IP addresses due to dynamic IP allocation, used by many Internet service providers. If both proxy servers and dynamic IP allocation are used, the problem becomes much more difficult to deal with.

Cookies Another option for data collection is cookies. They are small pieces of information carried between server and client [16]. Usually, they are used for tracking of users by web servers. The server sends some unique token to the client and every time the client requests some pages from the server, it sends that token back to server which is used to identify the client.

The use of cookies needs collaboration with users. However, cookies come with privacy issues and the most modern web browsers allow users to decide whether to enable or disable them. Therefore, some users may disable cookies which makes it impossible to use them for data collection.

Web Bugs A web bug is a small and usually 1×1 pixel transparent image file embedded in HTML file for tracking and getting information about client [17]. Since that image is usually transparent and small, it is not seen by the client. When the client requests the HTML file, it also requests that image file. That image file may be stored in another server and by requesting that image file, information about client (IP address, request time, web browser type, etc.) is also sent to that server.

Web bugs are useful to some extent even when browser's cookies are turned off. Similar to cookies, web bugs have privacy implications, especially when used in emails. Web bugs can be used by spammers to validate email addresses.

Explicit User Input It may be useful for some applications to use explicit user input. Data about users can be collected through forms. However, users do not want to bother if it is not very important for application. On the other hand, user supplied data may not be reliable [1].

2.5.2 Client-side Log Collection

The second option is to collect data in the client-side. This method has several advantages over server-side log collection. Unlike server-side collection methods, client-side collection does not need to deal with caching or session identification problems [8]. On the other hand, it has some disadvantages. Convincing users to allow the collection of his/her usage data is difficult [8]. People may easily feel disturbed by being monitored and not want to allow data collection.

Javascript Applications Javascript programs can be embedded in web pages and when they are executed on client-side, they gather information directly about client, time and duration of visits of pages etc. It is possible to disable javasript applications for privacy and security concerns, so user cooperation is essential.

Browser addons Data about browsing behaviors of users can be gathered using browser addons. Similar older technique were also used to modify web browser. Cunha et al. modified Mosaic web browser to collect data in [18]. It is not practical to modify modern web browsers, even if the source code is freely available. Another issue is to convince users to install and use the modified version. Modern way of doing this to install browser addons which collect data about clients. Again, cooperation of users is required.

2.5.3 Intermediary Data

Proxy Servers A proxy server is a software system that acts as intermediary between client browsers and servers. Proxy servers are used for security, caching and filtering purposes. Access logs of proxy servers are used as data source.

Packet Sniffers A packet sniffer is a software system or a hardware device that monitors TCP/IP packets on a network. Packet sniffers can be used for real-time data analysis, but in case of a problem, all data may be lost forever, since data are collected in real time and not logged [1].

2.6 Data Preprocessing

In this study, server access logs are used, so preprocessing steps given in following subsections use server access logs as input.

2.6.1 Data Fusion

For some applications, the web site to be analyzed may be large, or multiple related web sites may be used for analysis of user behaviors. Therefore, log files may come from different sources. For efficient analysis of log files, they must be merged accordingly.

2.6.2 Data Cleaning

Before starting analysis of logs, irrelevant log items that are not useful for analysis, should be removed. Consider the following simple scenario. There are image files I_1 and I_2 embedded in page A, and image files I_3 in page B. When the user requests page A and then page B, the navigation sequence would look like $A \rightarrow I_1 \rightarrow I_2 \rightarrow B \rightarrow I_3$. Image files do not provide information about the access sequence, unless image processing is done to extract content information from images. A simple heuristic is to remove logs with requests for specific file types. For instance, requests for image files with filename suffixes `.jpeg`, `.png` and `.gif` can be removed [6].

Web usage mining aims to extract behavior patterns of users that navigate through web pages. It is desirable to distinguish navigations of users from robots (also known as spiders and crawlers). Robots are software programs that traverse web pages using hyperlink structure to retrieve information [19]. There are some heuristics to identify log items that belong to robots. According to the Robots Exclusion Standard [20], the owner of web site `www.example.com` can place a text file called `robots.txt` to give instructions to robots to allow or prevent certain parts of the site to be accessed. According to this standard, whenever a robot visits `www.example.com`, it first checks the address `www.example.com/robots.txt` to get instructions of web site owner. Therefore, the request for `www.example.com/robots.txt` in access logs can be used to detect robots accessing to the web site. All IP addresses accessed to `robots.txt` file are considered as web robots and all log items with these IP addresses are removed from server logs.

For detection of robots, user agent field in Common or Extended log formats can be used. Cooperative web robots must declare their identity in user agent field [19], however if a robot attempts to hide its identity, it is not possible to detect it using user agent field. Another way to detect robots is to look for IP addresses [19]. There are many lists of robots and their IP addresses. By checking IP addresses of log items, robots can be detected. However, these lists may be incomplete and IP addresses of robots may change over time. Robots usually use specific strategies to retrieve pages on a web site, such as breadth-first search. These navigation patterns can be modeled and used to classify clients as user or robot [19].

Some unnecessary data fields such as number of bytes transferred, version of used HTTP protocol, etc. can be removed from log file [12].

2.6.3 Page View Identification

A page view is defined as a set of page files that contribute to a single display in the browser [2]. When the user clicks a link to web page P, n frames and m graphics are loaded into the browser. These n frames and m graphics form P [8]. These n HTML frames and m graphics would appear in log file. Even if graphics files are cleaned in data cleaning step, page P is still seen as request for P and requests for each of n HTML files. Instead of $n + 1$ different files, they must be combined as a page view and considered as one item.

2.6.4 User Identification

Since the ultimate goal is to extract navigation patterns of users, the user identification is a critical step in preprocessing process and it is difficult in systems without authentication mechanisms. There are some heuristics developed to identify users, but every method has some disadvantages.

The simplest heuristic is to consider each IP address as a different user. Several web usage mining systems adopt this approach [1]. Due to the use of proxy servers, a single IP address may be used by multiple users. On the other hand, in a network with dynamic IP allocation, IP address of a user may change over time, so a client may use multiple IP addresses.

Cooley et al. propose two different heuristics in [21]. The first one uses IP addresses together with agent field in server logs. If the IP addresses of two log items are same but agent fields are different, this heuristic assumes that they belong to different users.

The second heuristic proposed in [21] uses web site topology to identify users. For example, if there are requests for page A and then page B from the same IP address, and if there is no link from page A to B, it is assumed that requests for pages A and B are from different users. This heuristic may produce inaccurate identifications if there is a browser or proxy level cache mechanism. For instance, if a user has the browsing path $A \rightarrow B \rightarrow C$ and if B is cached in some way, the browsing path would look like $A \rightarrow C$ in server log file. A system that adopts this heuristic would assign requests for A and C to different users.

As another option, cookies can be used to identify users. When a user visits a web page, server sends a unique id with the page requested. After first request, every time the user requests a page from the server, it sends the id along with the request. Server receives unique id of client uses it to identify user. However, as mentioned in Section 2.5.1, they may not be preferred and disabled by users due to privacy concerns.

In our study, the pair of IP address and agent field is used to identify users. With this heuristic, the problem of single IP address/multiple users is solved to some extent. The second problem, multiple IP addresses/single user problem is less critical. Most of the time, sessions are not so long and IP address of a client is less likely to change during that time.

2.6.5 Session Construction

Session identification is another important step among preprocessing steps and it is an issue too without authentication mechanisms. A user session is the click-stream of page views for a user on the Web [8]. Basically, session construction is the partition of log item sets of a user into subsets, such that elements in a subset should be related and their access times should be close to each other. In a typical log file, there may be log items belonging to different days, even different weeks. Such items should belong to different sessions. On the other hand, access logs for not related URLs should be considered as members of different sessions, even if their access times are close to each other. There are different session construction heuristics developed to handle these cases [12, 22, 23]. They can be time or structure oriented.

Time-oriented ones generally use time threshold t and if visiting time difference of two pages are greater than t, they belong to different sessions. Although threshold t is highly dependent to content of the site, $t = 30$ minutes is used as default [1, 21]. Table 2.2 gives an example of session construction using time threshold $t = 30$ minutes. Since the time difference between access to `C` and access to `E` is greater than 30 minutes, they belong to different sessions.

Table 2.2: Time-oriented session construction example

	IP address	time	URL	referrer
user 1	1.2.3.4	[0:00]	A	-
	1.2.3.4	[2:17]	B	A
	1.2.3.4	[15:32]	C	B
user 2	1.2.3.4	[47:11]	E	A
	1.2.3.4	[49:39]	F	E

Table 2.3: Time-oriented session construction example when caching mechanism is used. Log items between two dashed lines are retrieved from the cache and do not appear in server log file.

IP address	time	URL	referrer
1.2.3.4	[0:00]	A	-
1.2.3.4	[2:17]	B	A
1.2.3.4	[12:32]	C	B
1.2.3.4	[17:11]	A	C
1.2.3.4	[29:39]	B	A
1.2.3.4	[45:08]	D	B
1.2.3.4	[49:44]	E	D

If browser or proxy server caching is used, incorrect sessions may be constructed. Table 2.3 shows an example. Log items between dashed lines are not server logs. They are loaded from the cache, in other words they are not requested from the server, so these requests are not recorded in server access logs. The correct browsing history is A → B → C → A → B → D → E. However, due to caching mechanism, the browsing history in log files would look like A → B → C → D → E. In that case, since the time difference between request for C and the request for D is greater than threshold, there would be two sessions, although there is one.

As the second heuristic, hyperlink structure of web sites is used to construct sessions. Existence of hyperlinks between two consecutive log items or referer fields of server logs can be used. For example, in Table 2.4, referer field of the last log item is empty which means it was not accessed from page C. Therefore, third and last log items should belong to different sessions.

Similar to time-oriented method, caching may cause problems for structure-oriented method. Consider the following scenario: Let the original browsing path be A → B

Table 2.4: Structure-oriented session construction example

IP address	time	URL	referrer
1.2.3.4	[0:00]	A	-
1.2.3.4	[0:15]	B	A
1.2.3.4	[8:37]	C	B
1.2.3.4	[15:43]	E	-

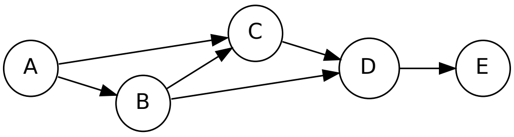


Figure 2.1: Sample web site hyperlink structure

→ C and let B be retrieved from cache somehow. In the log file, browsing path would look like A → C and if there is no hyperlink between A and C, they would be assigned to different sessions.

2.6.6 Path Completion

Because of caching mechanism, some files are retrieved from the cache and these requests are not logged in server-side. Path completion is used to recover missing items in the log file. It is usually performed after session construction [12]. To complete missing requests, web site structure and referer field of log file are used [21].

A simple example is given in Figure 2.1. If the original browsing path is A → B → C → D → E and if D is missing in log file (i.e. the path looks like A → B → C → E), page D can be recovered using link structure. There is no direct way to access from C to E, there is only one link going outside from C and only one link coming to E. D is the only possible page, so it is assumed that it is actually retrieved but does not appear in log file due to caching.

2.7 Clustering for Pattern Discovery

After preprocessing, different methods can be applied to extract meaningful and useful patterns from data. These methods are adopted from different fields, such as statistics, machine learning and data mining. This section describes clustering, the most commonly used one for pattern discovery on web mining area.

Clustering is to divide a set of objects into meaningful groups. The goal is to partition objects such that similar objects will be in same groups while different ones will be in different groups.

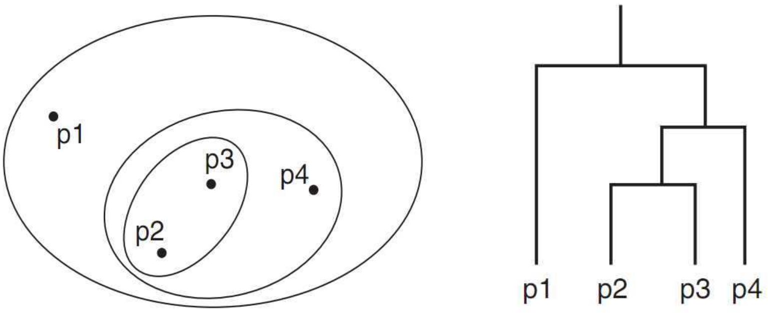
Cluster analysis is used in wide range of fields such as biology, psychology, medicine, business, image processing, information retrieval, data mining and web usage mining [24, 25].

2.7.1 Clustering Types

There are many different types of clusterings [24] and they are described below.

- **Hiearchical versus Partitional** A partitional clustering is divison of set of objects into non-overlapping clusters. In a hierarchical clustering, clusters contain subclusters. Usually, the clustering is represented with a tree-like diagram called dendrogram. An example of hierarchical clustering and its dendrogram are given in Figure 2.2.

  The hierarchical clustering can be built with agglomerative (bottom up) or divisive (top down) strategies. In agglomerative approach, at the beginning, each instance is a cluster and at each step, a pair of these clusters are merged. Merge operation is repeated until one final cluster having all instances is produced. Divisive approach starts with one cluster having all objects and divides it into subclusters. At each step, it divides a cluster that contains multiple objects into subclusters. The process is repeated until there is no cluster to divide. For both agglomerative and divisive approaches, all intermediate clusters are recorded and used for building dendrogram.



(a) (b)

Figure 2.2: Sample hierarchical clustering and its dendrogram

- **Exclusive versus Overlapping versus Fuzzy** Depending the type of application, a clustering may be exclusive, overlapping (non-exclusive) or fuzzy. In an exclusive clustering, an object belongs to only one cluster, while it belongs to multiple clusters at the same time in overlapping clusterings. In a fuzzy clustering, the degree of membership of an object to a cluster can be defined with membership weight. Usually the sum of membership weights of an object is 1.0.

- **Complete versus Partial** In a complete clustering, every object is assigned to a cluster while there may be objects that are not assigned to any cluster, in a partial clustering.

2.7.2 Distance Measures

One of the critical steps is to select distance measure. In following subsections there are several distance/similarity measures explained.

Eucledian Distance The most common measure used in clustering is Eucledian distance. Each instance X is represented as a vector of features $X = (X_1, X_2, \ldots, X_n)$. If $X = (X_1, X_2, \ldots, X_n)$ and $Y = (Y_1, Y_2, \ldots, Y_n)$ are two elements in n dimensional Eucledian space, distance between them is formally defined as

$$d(X,Y) = \left(\sum_{i=1}^{n} (X_i - Y_i)^2 \right)^{1/2} \tag{2.1}$$

Generalization of Eucledian distance is called Minkowski distance where distance between two points X and Y is defined as

$$d(X,Y) = \left(\sum_{i=1}^{n} |X_i - Y_i|^p \right)^{1/p} \tag{2.2}$$

In Eucledian distance $p = 2$. Minkowski distance is also widely used with $p = 1$, which is known as Manhattan distance.

Cosine Similarity Another measure of similarity is cosine similarity which is the most common measure of document similarity [24]. The cosine similarity of two vectors X and Y is defined as

$$\cos(X,Y) = \frac{X \cdot Y}{||X||\ ||Y||} \tag{2.3}$$

where $\cdot$ represents the vector dot product, $X \cdot Y = \sum_{i=1}^{n} X_i Y_i$ and $||X||$ is the length of vector X, $||X|| = \sqrt{\sum_{i=1}^{n} X_i^2}$.

Jaccard Coefficient Given two objects A and B with n binary attributes (i.e. $A = \langle a_1, \ldots, a_n \rangle$, $B = \langle b_1, \ldots, b_n \rangle$ and $a_i, b_j \in \{0,1\}$, $(1 \leq i, j \leq n)$), the Jaccard coefficient [24] is the measure of attributes shared by A and B. The following quantiles are used to compute similarity between A and B.

- M_{01} is the number of attributes where the attribute of A is 1 and the attribute of B is 0.
- M_{10} is the number of attributes where the attribute of A is 0 and the attribute of B is 1.
- M_{00} is the number of attributes where A and B both have value 0.
- M_{11} is the number of attributes where A and B both have value 1.

So, $M_{00} + M_{01} + M_{10} + M_{11} = n$ and Jaccard coefficient is

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{00}} \tag{2.4}$$

Tanimoto similarity is the extension to Jaccard Coefficient for measuring similarities between objects with not binary features [24]. It is also known as Extended Jaccard Coefficient.

$$T(A,B) = \frac{A \cdot B}{||A||^2 + ||B||^2 - A \cdot B} \tag{2.5}$$

Correlation Coefficient Pearson Correlation Coefficient measures the similarity between two objects $A = \langle A_1, \ldots, A_n \rangle$ and $B = \langle B_1, \ldots, B_n \rangle$. It is defined as

$$\frac{\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \bar{A})^2}\sqrt{\sum_{i=1}^{n}(B_i - \bar{B})^2}} \tag{2.6}$$

where $\bar{A} = \frac{1}{n}\sum_{i=1}^{n} A_i$.

2.7.3 Distances Between Two Clusters

Hierarchical clustering algorithms use the distance between two clusters. Single link, complete link and average link distance are mainly used measures to measure distance between two clusters. Single link distance between two clusters is the minimum distance between any two points in two different clusters. Complete link distance between two clusters is the maximum distance between any two data points in two different clusters. Third one, average link between two clusters is the average pairwise distance among all points in different clusters. Formal definitions of single, complete and average link distance between two clusters A and B are given in equations 2.7, 2.8 and 2.9 where $|A|$ is the number of elements in cluster A.

$$d_{\text{single}}(A,B) = \min_{a \in A,\ b \in B} d(a,b) \tag{2.7}$$

$$d_{\text{complete}}(A,B) = \max_{a \in A,\ b \in B} d(a,b) \tag{2.8}$$

$$d_{\text{average}}(A,B) = \frac{\sum_{a \in A,\ b \in B} d(a,b)}{|A|\ |B|} \tag{2.9}$$

2.7.4 Clustering Algorithms

***k*-means** is the most common clustering algorithm used in machine learning and data mining [24, 26]. The algorithm partitions data into k clusters such that each object is the member of the cluster with closest centroid. The centroid of a cluster is usually the arithmetic mean of member objects. k, the number of final clusters is given as input. The algorithm 1 gives the steps of k-means algorithm.

Algorithm 1 k-means clustering algorithm

```
Select $k$ points as initial centroids of clusters.
repeat
  Assign every data point to the cluster with nearest centroid.
  Recalculate the centroid of each cluster using member data points of it.
until No change in centroids.
```

The number of clusters k is not determined automatically and should be given as input. Another drawback of k-means clustering algorithm is choosing initial centroids. Different selection of initial centroids may yield different clusters at the end. Therefore, poor selection of initial centroids may produce poor results.

There are several modifications to original k-means algorithm in literature. In fuzzy c-means clustering [27], each object belongs to more than one cluster with different degrees of membership. Another extension for k-means is x-means algorithm [28]. x-means algorithm estimates the number of clusters without any user input.

Hierarchical algorithm is another widely used clustering algorithm. As explained above, agglomerative and divisive approaches are used to build a hierarchical clustering. The pseudocode of agglomerative approach is given in Algorithm 2. In divisive approach, at each step, a cluster is chosen to be split based on some criteria. The process is repeated until all clusters have single object. To measure distance between clusters, single, complete or average link measures can be used.

Algorithm 2 Agglomerative hierarchical clustering algorithm

```
repeat
    Merge the closest two clusters.
    Calculate the distance between merged and remaining clusters.
until Only one cluster remains.
```

DBSCAN (Density Based Spatial Clustering of Applications with Noise) [29] is another widely-used partitional clustering algorithm. Unlike k-means clustering algorithm, DBSCAN does not require k as input, but it requires some other parameters, which yields the same problem. Poor choice of these parameters may yield to bad clustering at the end.

2.7.5 Cluster Evaluation

One of the most widely used cluster evaluation method is to calculate inter-cluster and intra-cluster similarity values. The intra-cluster similarity is the measure of how tight a cluster is, in other words, how much items in a cluster are close to each other. In a good clustering, average intra-cluster similarity is high. The average intra-cluster similarity of a clustering $\mathcal{C} = \{C_1, \ldots C_n\}$ is

$$IS(\mathcal{C}) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{d,d' \in C_i} s(d, d') \right) \tag{2.10}$$

where $s(d, d')$ is the similarity between items d and d'.

The inter-cluster similarity is the measure of how well clusters are separated from each other. In a good clustering, average inter-cluster similarity is low, because clusters are well separated from each other. The average inter-cluster similarity for a clustering $\mathcal{C} = \{C_1, \ldots C_n\}$ is

$$ES(\mathcal{C}) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} S(C_i, C_j) \tag{2.11}$$

where $S(C_i, C_j)$ is the average pairwise similarity between items in clusters C_i and C_j.

$$S(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{d_i \in C_i, d_j \in C_j} s(d_i, d_j) \tag{2.12}$$

where $s(d_i, d_j)$ is the similarity between items d_i and d_j.

In addition to inter/intra-cluster similarity measures, Halkidi et al. survey clustering validation techniques and give comparison of widely known clustering algorithms [30].

2.8 Other Pattern Discovery Methods

In this section, pattern discovery methods other than clustering are described. Similar to clustering method, they are usually unsupervised, since labeling instances of large datasets is not easy [1].

2.8.1 Association Rules

Association rules are used to represent associations between items in a dataset [24]. A rule $X \Rightarrow Y$ means that there is a strong relationship between the occurrence of X and Y item sets (i.e. collections of one or more items). Two metrics, support and confidence are used to evaluate the strength of association rules. Support of a rule $X \Rightarrow Y$, $S(X \Rightarrow Y)$ is the fraction of transactions that contain both X and Y. Confidence of $X \Rightarrow Y$, $C(X \Rightarrow Y)$ is the measure of how often Y appears in transactions that contain X. Formally, support and confidence are defined as follows.

$$S(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{2.13}$$

$$C(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{2.14}$$

where $\sigma(X)$ is the support count, the number of transactions that contain X. N is the total number of transactions in the dataset. Association rule mining is the task of finding all rules with support s and confidence c such that $s \geq minsup$ and $c \geq minconf$, where *minsup* is support threshold and *minconf* is the confidence threshold. The most popular algorithm used to discover association rules is Apriori algorithm [31].

In the context of web usage mining, item sets are sets of web pages accessed and association rule mining is used to discover the set of web pages accessed together in a user session. Given a set of web pages accessed by the user, other frequently co-occurred pages may be recommended to the user. Another possible use is to cache associated pages that are not accessed yet.

2.8.2 Sequential Patterns

Sequential pattern mining is the task of finding frequent sequences of events. Different from association rule mining, it includes information of time order between events. Extended association rule mining algorithms are used for sequential patterns. Some algorithms based on Apriori algorithm [31] are AprioriAll, AprioriSome, DynamicSome [32], GSP (Generalized Sequential Pattern) [33] and SPADE (Sequential PAttern Discovery using Equivalence classes) [34]. FreeSpan [35] and PrefixSpan [36] are based on data projection method.

When applied to web usage mining, discovered sequential patterns can be used to predict next web page to be visited by the user.

2.8.3 Classification

Classification is the assignment of items to predefined classes [24]. Classification is a supervised technique which means a training set consisting of instances and their class labels is available. Decision tree induction, neural networks, Bayesian classifiers and support vector machines are mainly used methods.

In web usage mining, classification methods are used to assign users to predefined classes based on their navigations. The use of classification for web usage mining is limited when compared with unsupervised methods. For supervised learning, a set of preclassified instances are required and manual classification of large number of instances is not an easy task [1].

CHAPTER 3

MINING FREQUENT PATTERNS FROM WEBSITE LOGS USING ONTOLOGY

In this study, web navigation information of users is integrated with the set of concepts defining web pages. Each web page in the domain is described with some concepts in the taxonomy based on its content. Although the boundary between content, structure and usage mining is not so clear [11, 13], the method of this study can be classified as the hybrid of web content and usage mining.

In this study, clustering methods are applied on web usage data to discover frequent patterns. Given a set of sessions, sessions are clustered to find meaningful partition with the aim of maximizing intra-cluster similarity while minimizing inter-cluster similarity. Each session is a sequence of web pages and each web page is represented with a set of concepts from the taxonomy defined.

Figure 3.1 gives the overview of the method. The first step is the collection of web server logs and preprocessing of them. In preprocessing phase, sessions are constructed from logs and using manually defined taxonomy, each web page is mapped to a set of concepts from the taxonomy.

After preprocessing step, a series of similarity measures are needed to cluster sessions. To measure similarity of two session $S_i = \langle P_1^{(i)}, \ldots, P_{n_i}^{(i)} \rangle$ and $S_j = \langle P_1^{(j)}, \ldots, P_{n_j}^{(j)} \rangle$, the similarity among web pages of these two sessions is measured. It is the similarity between two web pages $P_a^{(i)}$ and $P_b^{(j)}$ for all pairs of a and b such that $1 \leq a \leq n_i$ and $1 \leq b \leq n_j$. It is measured using the similarity between two sets of concepts $C_a^{(i)}$ and $C_b^{(j)}$ defining $P_a^{(i)}$ and $P_b^{(j)}$, respectively. To measure similarity between sets of concepts similarity between concepts is required. Measures of similarity of two concepts, measure of similarity of two web pages (two sets of concepts) are given in following subsections 3.4 and 3.5. Method used to measure similarity between two sessions is also given in subsection 3.6.

After definition of similarity measures, web sessions are partitioned into clusters. To assign a new instance to one of the clusters, it can be compared with all clusters and the closest cluster should be selected.

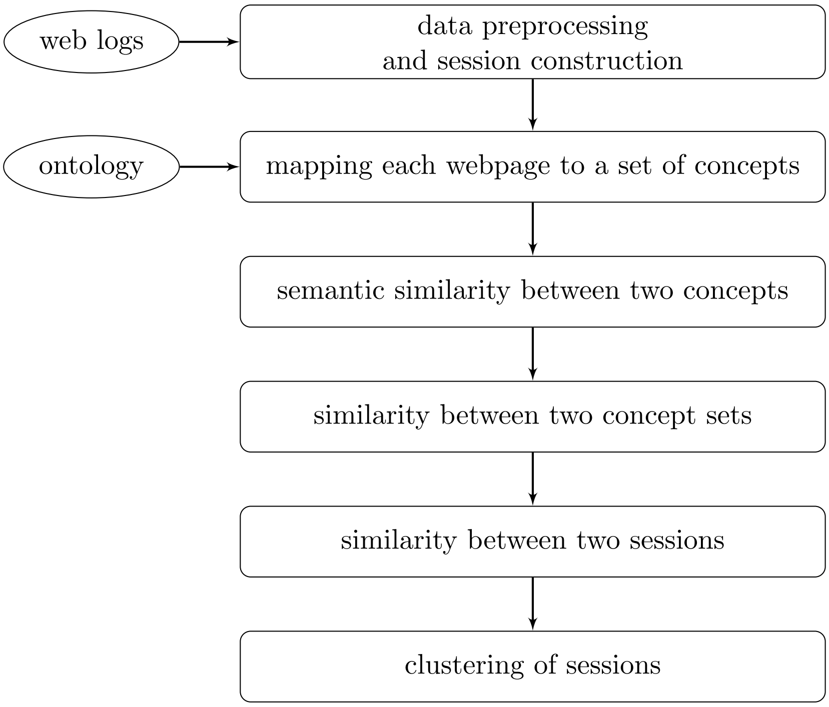

web logs
data preprocessing and session construction
ontology
mapping each webpage to a set of concepts
semantic similarity between two concepts
similarity between two concept sets
similarity between two sessions
clustering of sessions

Figure 3.1: Overview of the method

3.1 Data Preprocessing and Session Construction

To train and test our system, we use access logs of a web server. Since studying on entire web is practically impossible, we restrict our domain to Middle East Technical University (METU) Computer Engineering Department website[1]. Preprocessing steps usually followed in web usage mining are described in Section 2.6. In this study, we

[1] `http://www.ceng.metu.edu.tr`

usually use simple heuristics to process data which are briefly explained below.

Data Fusion Since we use only one data source for our logs, no data fusion method is used in this study.

Data Cleaning From web logs, we remove log items that are not useful for extraction of navigation patterns. We do not process multimedia files, archive files and external documents. Therefore, we remove logs of requests to these items. To identify these items, we use suffixes of filenames requested. Multimedia files (with extensions `.png`, `.jpg`, `.mp3`, `.avi`, `.gif`, etc.), archive files (with extensions `.rar`, `.tar`, `.zip`, etc) and external document files (with extensions `.pdf`, `.doc`, `.ppt`, etc.) are removed from logs.

For detection and removal of logs belong to web crawlers, we use a simple heuristic. We identify all IPs accessed to `robots.txt` as web crawlers and remove all access logs belonging to these IPs from the dataset.

Page View Identification As explained in section 2.6.3, multiple files of text and graphics can be loaded in the same view. After cleaning graphics, these text files are combined as a page view. All page requests from a single IP address within the same second are considered as members of the same page view. Sample from server log of `www.ceng.metu.edu.tr` is given in Table 3.1. The page `courses/ceng436` consists of two frames: `csToolbar.html` and `csMain.html`. `csToolbar.html` has six GIF image files and `csMain.html` has no image files embedded into it. When the client `1.2.3.4` requests `/courses/ceng436`, two `HTML` files and six GIF image files are requested too. Assuming GIF image files are removed in data cleaning step, there would be three log items with same time fields: `/course/ceng436`, `/course/ceng436/csToolbar.html` and `/course/ceng436/csMain.html`. Since time fields of all three of them are same, they are considered in the same page view. The time field of the last URL (`/courses/ceng436/lectures/index.html`) is different, therefore it belongs to a different page view.

Table 3.1: Sample from server access log. Only IP address, time and requested URL fields are given. IP address is changed for privacy protection.

IP	time	URL
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/csToolbar.html
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/csMain.html
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/smLogoBlack.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbChat.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbLinks.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbSyllabus.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbLectures.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbAssign.gif
1.2.3.4	[13/Feb/2011:20:09:29 +0200]	/courses/ceng436/img/tbAnnoun.gif
1.2.3.4	[13/Feb/2011:20:10:27 +0200]	/courses/ceng436/lectures/index.html

User Identification We apply a simple heuristic to identify users. We use IP address and agent fields of logs together and assign two different logs to the same user if their IP address and agent fields are same.

Session Construction For session construction, we adopt time-oriented heuristic. We consider two consecutive visits of a user in the same session if access time difference between them is not more than some threshold t. In our study we use $t = 30$ minutes.

3.2 Building Taxonomy

In this study, instead of using the content (usually in text format) of web pages directly, each web page is described with a set of keywords. To assess similarity of web pages, similarity of these keywords are used. Taxonomy of keywords (concepts) is built to model properties of these concepts and relationships among them. ISA ("is a") hierarchy (taxonomy) of the example computer science department ontology from SHOE (Simple HTML Ontology Extensions) project [37]. The original ontology is available at[2]. The modified taxonomy used in this study is also available at[3].

[2] http://www.cs.umd.edu/projects/plus/SHOE/cs.html

[3] http://www.ceng.metu.edu.tr/~sefa/msthesis/ontology.dat

Table 3.2: Some concepts from the ontology and keyword sets describing them

Concept	Associated keyword sets
`ResearchLaboratory`	{research, laboratory}
`Bioinformatics`	{bioinformatics}
`ImageProcessing`	{image, processing}, {pattern, recognition}
`OperatingSystems`	{operating, system}, {process}, {thread}, {deadlock}, {memory, management}
`UndergraduateStudent`	{undergraduate, student}
`GraduateStudent`	{graduate, student}

3.3 Web Page to Concepts Mapping

For measure of similarity between two web pages, we map each web page to a set of concepts in the ontology defined. Each concept in the taxonomy is associated with some keywords. A set of concepts and keywords describing them are given in Table 3.2.

Each web page is represented as a bag of words. In other words, it is represented as unordered collection of words. A concept is in the mapping of a web page if all keywords in one of associated keyword sets appear in the web page. For example, to label a web page P with concept `ImageProcessing`, P should contain both words “image” and “processing”, or it should contain both “pattern” and “recognition”.

Example Let the set of concepts in Table 3.2 be all concepts that we use. The web page $P =$ `http://www.ceng.metu.edu.tr/∼tcan/` contains word “bioinformatics”, so it is tagged with concept `Bioinformatics`. The web page does not contains keywords “image” and “processing”. It contains “pattern”, but to use a concept with a web page, it should contain all keywords in an associated keyword set. The web page contains “pattern” but it does not contain the word “recognition”. Therefore it is not labeled with the keyword `ImageProcessing`. The web page contains both words “research” and “laboratory”, so it is labeled with the concept `ResearchLaboratory` too. It does not contain every word of any keyword set of other concepts. Therefore, the web page P is mapped to set of concepts $\{\texttt{ResearchLaboratory}, \texttt{Bioinformatics}\}$.

3.4 Measure of Similarity of Two Concepts

In this study, we map each web page to a set of concepts. To measure similarity of two web pages (two sets of concepts), we need pairwise similarity/distance measure for similarity/distance of two concepts. All of the used concepts are represented in a taxonomy and taxonomic relations of concepts are used to measure similarity/distance between them.

Several measures are described in this section. Some of them measure distance between concepts, others measure similarity. To avoid confusion, it should be noted that small distance means high similarity and vice versa. For agreement, all distance measures can be converted similarity or all similarity measures can be converted to distance measures. The similarity/distance measures given below were studied comparatively in [38],[39].

Some definitions used in definitions of similarity/distance measures are given below.

- $paths(x, y)$: set of paths between concepts x and y in the concept hierarchy.
- $len_e(p)$: length of path p (the number of edges that p uses).
- $len_n(p)$: length of path p (the number of nodes on path p).
- $mscs(x, y)$: the most specific common subsumer of x and y. It is the most specific concept in concept hierarchy that is superconcept of both x and y.
- $P(x)$: occurrence probability of concept x in the dataset.
- C: the set of all concepts.
- rt: the concept hierarchy root.
- D: the maximum hierarchy depth of concept hierarchy tree, $D = \max_{c \in C} \left(\max_{p \in paths(c,rt)} len_n(p) \right)$

Rada et al.'s Distance [40] It is simply the length of shortest path connecting two concepts in the concept hierarchy. Rada et al.'s distance between two concepts c_i

and c_j is defined as

$$dist_{\text{Rada}}(c_i, c_j) = \min_{p \in paths(c_i, c_j)} len(p) \tag{3.1}$$

Resnik's Similarity [41] Resnik proposed an information-theoretic similarity measure in [41]. The similarity between two concepts is defined as the information shared by these concepts, the information content of the most specific common subsumer. The information content of a concept c is quantified as $-\log p(c)$. Resnik's similarity of concepts c_i and c_j is defined as

$$sim_{\text{Resnik}}(c_i, c_j) = -\log P(mscs(c_i, c_j)) \tag{3.2}$$

If two concepts are so different, their most common subsumer concept will be more general (i.e closer to the root concept in concept hierarchy) where the negative logarithm of its occurrence probability will be small, and thus their similarity will be small.

Leacock and Chodorow's Similarity [42] Similar to Rada et al.'s distance given in Equation 3.1, this measure uses shortest path between concepts. The measure is normalized with the double of the maximum hierarchy depth D. Leacock-Chodorow's similarity of two concepts c_i and c_j is

$$sim_{\text{LC}}(c_i, c_j) = -\log \frac{\min\limits_{p \in paths(c_i, c_j)} len_n(p)}{2D} \tag{3.3}$$

Jiang and Conrath's Distance [43] Like Resnik's similarity, Jiang and Conrath have used occurrence probabilities of concepts, as well as taxonomic links between them. They have defined the distance between two concepts as the difference between the information content of their most common subsumer and the sum of information content of them.

$$dist_{\text{JC}}(c_i, c_j) = 2 * \log P(mscs(c_i, c_j)) - (\log P(c_i) + \log P(c_j)) \tag{3.4}$$

Lin's Similarity [44] It uses the same components with Jiang-Conrath's distance in Equation 3.4, but in the form of ratio. Lin's similarity of concepts c_i and c_j is defined as

$$sim_{\text{Lin}}(c_i, c_j) = \frac{2 * \log P(mscs(c_i, c_j))}{(\log P(c_i) + \log P(c_j))} \tag{3.5}$$

Example Consider sample concept taxonomy in Figure 3.2.

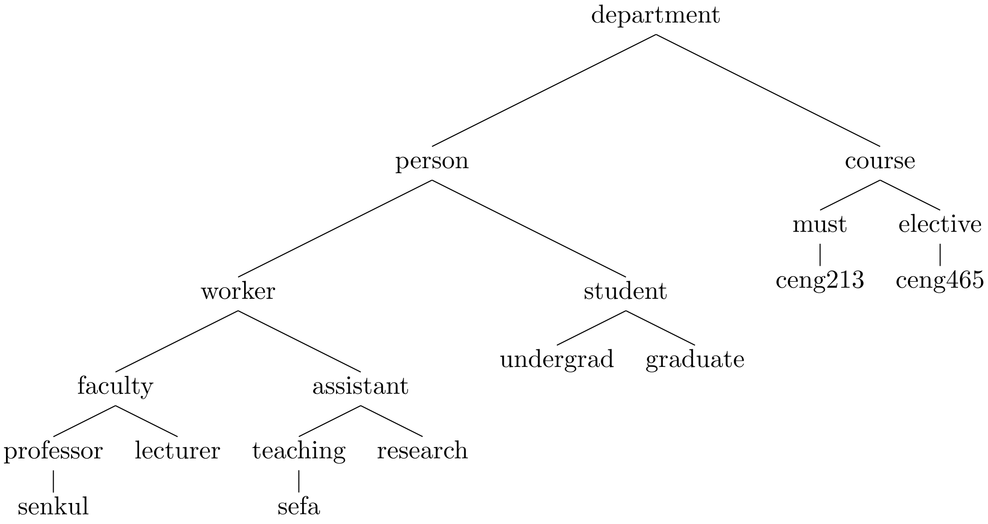


Figure 3.2: Sample concept taxonomy

Let c_1 = `professor`, c_2 = `teaching assistant` For each measure given above, the distance/similarity of c_1 and c_2 is as follows.

$$\begin{aligned}
dist_{\text{Rada}}(c_1, c_2) &= 4, \text{ (the path } \langle\text{professor,faculty,worker,assistant,teaching}\rangle) \\
sim_{\text{Resnik}}(c_1, c_2) &= -\log(p(\text{worker})), \text{ (worker is the mscs concept)} \\
sim_{\text{LC}}(c_1, c_2) &= -\log\frac{4}{2D}, \ (D = 5, \text{ the maximum hierarchy depth}) \\
dist_{\text{JC}}(c_1, c_2) &= 2 * \log p(\text{worker}) - (\log p(\text{professor}) + \log p(\text{teaching})) \\
dist_{\text{Lin}}(c_1, c_2) &= \frac{2 * \log P(\text{worker})}{(\log P(\text{professor}) + \log P(\text{teaching}))}
\end{aligned}$$

3.5 Measure of Similarity of Two Web Pages

The similarity measures used to calculate the similarity between two web pages are given in Section 3.5.1 below. Each web page is represented with a set of concepts and similarity measures between pairs of concepts are used to get the degree of similarity between two web pages.

To improve the similarity score between two web pages, the importance component is introduced and used together with the similarity component [45]. The importance component computes how important the similarity between two web pages and how much it should contribute to the overall similarity between two sessions containing these two web pages. It uses the fraction of time spent at these pages. Given two web pages P_i and P_j from sessions S_i and S_j respectively, let the similarity component be denoted by $\mathcal{S}'$ which is the similarity between two concept sets describing two web pages P_i and P_j. The importance component $\mathcal{S}''$ is given by

$$\mathcal{S}'' = \left(\frac{T(P_i)}{T(S_i)} \times \frac{T(P_j)}{T(S_j)} \right)^{1/2} \tag{3.6}$$

where $T(P_i)$ is the time spent on page P_i and $T(S_i)$ is the total time spent on session S_i. The total similarity between two web pages $P_i \in S_i$ and $P_j \in S_j$ is given by

$$\mathcal{S}(P_i, P_j) = \mathcal{S}' \times \mathcal{S}'' \tag{3.7}$$

The motivation for the importance component $\mathcal{S}''$ is that if two pages are semantically close to each other but fraction of time spent on these pages are small in overall sequences, then these two pages should not contribute to overall similarity so much. Small $T(P_i)/T(S_i)$ means page P_i is not an important element in session S_i. Maybe, user just visited page P_i to access another page where page P_i has a link to it. On the other hand large value of $T(P_i)/T(S_i)$ means page P_i is important in session S_i and should be used in the measurement of similarity of two sessions S_i and S_j.

3.5.1 The Similarity Component

Set similarity measures used in this study are reviewed in [39]. They compare two sets of concepts based on concept similarity/distance measures given in Section 3.4.

Hausdorff Distance It takes only the most distant objects into account. Given two sets of objects (i.e. concepts) A and B, Hausdorff distance is defined as

$$D_h(A,B) = \max\left(\max_{a\in A}(min\{d(a,b)|b\in B\}), \max_{b\in B}(min\{d(a,b)|a\in A\})\right) \tag{3.8}$$

where $d(a,b)$ is the distance between two concepts a and b where $a \in A$ and $b \in B$. As pointed in [46], it is very sensitive to extreme points. Figure 3.3 shows such an example.

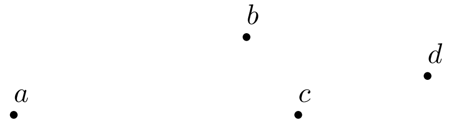


Figure 3.3: According to Hausdorff distance, sets $\{a,b,c\}$ and $\{a\}$ are equally distant from the set $\{d\}$.

Sum of Minimum Distances The sum of minimum distance between two sets of objects A and B is defined as

$$D_{smd}(A,B) = \frac{1}{2}\left(\sum_{a\in A}\left(\min_{b\in B} d(a,b)\right) + \sum_{b\in B}\left((\min_{a\in A} d(a,b)\right)\right) \tag{3.9}$$

Surjection Distance [46] A relation $\eta \subseteq A \times B$ is a surjection from set A onto B if $\forall b \in B, \exists a \in A : (a,b) \in \eta$. The surjection distance measures the distance between two sets A and B by using surjections (η) from larger set to smaller one.

$$D_s(A,B) = \min_{\eta} \sum_{(e_1,e_2)\in\eta} d(e_1,e_2) \tag{3.10}$$

where the surjection with minimum total distance is selected over all surjections from larger set to the smaller one.

Link Distance [46] A linking between two sets A and B is a relation $R \subseteq A \times B$ such that $\forall a \in A,\ \exists b \in B : (a,b) \in R$ and $\forall b \in B,\ \exists a \in A : (a,b) \in R$. In other words, every object of set A is associated with an object in B and vice versa. The link distance of sets A and B is defined as

$$D_l(A,B) = \min_R \sum_{(a,b)\in R} d(a,b) \tag{3.11}$$

Matchings A matching is the set of matches between elements in set A and elements in set B such that each object in A is associated with at most one object in B and vice versa. A maximal matching is a matching when no more associations can be added. Assuming $m^m(A,B)$ are all maximal matchings between A and B, the matching distance [47] is defined as

$$D_m = (A,B) = \min_{r\in m^m(A,B)} d(r,A,B) \tag{3.12}$$

Average Linkage Based Similarity The average linkage based similarity of two sets A and B is the average of similarities of all possible pair of objects a and b from sets A and B.

$$sim_{al}(A,B) = \frac{1}{n_A n_B} \sum_{a\in A} \sum_{b\in B} sim(a,b) \tag{3.13}$$

where n_X is the number of objects in set X.

Single Linkage Based Similarity The single linkage based similarity of two sets A and B is the maximum similarity of any pair of objects a and b from sets A and B.

$$sim_{sl}(A,B) = \max_{a\in A,\ b\in B} sim(a,b) \tag{3.14}$$

Table 3.3: Rada et al.'s distance of the sample of concepts.

	sefa	graduate	ceng465
senkul	6	6	8
ceng213	8	6	4

Example Consider two web pages P_1 and P_2 where they are mapped to concept sets $C_1 = \{\texttt{senkul}, \texttt{ceng213}\}$ and $C_2 = \{\texttt{sefa}, \texttt{graduate}, \texttt{ceng465}\}$ respectively. Using Rada et al.'s distance the distances between concepts are given in Table 3.3.

In this study, average linkage based similarity is used. First, distances are converted to similarities. For example, the distance between concepts `ceng213` and `ceng465` is 4, so the similarity is 1/4. The similarity between sets C_1 and C_2 is

$$\begin{aligned} sim(C_1, C_2) &= \frac{1}{n_A n_B} \sum_{a \in A} \sum_{b \in B} sim(a, b) \\ &= \frac{1}{2 \times 3} (1/6 + 1/6 + 1/8 + 1/8 + 1/6 + 1/4) \\ &= 0.166 \end{aligned}$$

3.6 Measure of Similarity of Two Sessions

To measure similarity between two sessions (i.e. ordered set of web pages), sequence alignment methods were used in [5, 48, 49]. In this work, we also use well known Needleman-Wunsch algorithm [50].

Needleman-Wunsch Algorithm Needleman-Wunsch algorithm [50] is an example of dynamic programming. It is commonly used in bioinformatics for global alignment of protein or nucleotide sequences.

The goodness of alignment is measured with alignment score, the sum of scores of each pair of aligned items. Gaps can be inserted to increase alignment score, with a determined gap penalty. As an example, let a and b be two nucleotide sequences where $a = \langle \texttt{A}, \texttt{T}, \texttt{G}, \texttt{C}, \texttt{A} \rangle$, and $b = \langle \texttt{A}, \texttt{C}, \texttt{T}, \texttt{G}, \texttt{T}, \texttt{G} \rangle$. With match score $S(x, x) = +4$, mismatch score $S(x, y) = -2$, $(x \neq y)$ and gap penalty $S(x, -) = -1$, the score of alignment

```
A-TGCA
ACTGTG
```

is $S(\mathtt{A},\mathtt{A}) + S(\mathtt{-},\mathtt{C}) + S(\mathtt{T},\mathtt{T}) + S(\mathtt{G},\mathtt{G}) + S(\mathtt{C},\mathtt{T}) + S(\mathtt{A},\mathtt{G}) = 4 - 1 + 4 + 4 - 2 - 2 = 7$.

With dynamic programming approach, Needleman-Wunsch algorithm solves the problem by breaking it into subproblems. The optimal alignment of two sequence is built by using optimal alignments of subsequences. Let $a = \langle a_1, \ldots, a_m \rangle$ and $b = \langle b_1, \ldots, b_n \rangle$ be two sequences to be aligned. A matrix (two dimensional array) M of size $(m+1) \times (n+1)$ is allocated for alignment scores. $M(i,j)$ is the score of optimal alignment of subsequences $a' = \langle a_1, \ldots, a_i \rangle$ and $b' = \langle b_1, \ldots, b_j \rangle$. For each position, $M(i,j)$ is defined as follows

$$M(i,0) = d \cdot i \tag{3.15}$$

$$M(0,j) = d \cdot j \tag{3.16}$$

$$M(i,j) = \max \begin{cases} M(i-1,j) + d \\ M(i,j-1) + d \\ M(i-1,j-1) + S(a_i, b_j) \end{cases} \tag{3.17}$$

where d is the gap penalty and $S(\cdot,\cdot)$ is the match/mismatch score of a pair of sequence items. The pseudocode 3 gives steps of filling matrix M.

When matrix M is computed, $M(m,n)$ is the score of optimal alignment of sequences a and b. In addition to maximum score, the optimal alignment itself can be found using matrix M. Since, we need only maximum score in our study, the algorithm to find optimum alignment is not given here and can be found in [50].

We use Needleman-Wunsch dynamic programming algorithm to measure the similarity between two sessions $S_i = \langle P_i^1, \ldots, P_i^n \rangle$ and $S_j = \langle P_j^1, \ldots, P_j^m \rangle$. P_a^b is the bth visited web page in session a and it is represented with a set of concepts from the ontology defined. Instead of nucleotide or protein sequences, we align sequences of web pages (i.e. sequences of concept sets). Therefore, instead of match/mismatch score, we use concept set similarity/distance score of two web pages to calculate alignment score of

Algorithm 3 Optimal alignment score of sequences a and b

```
for $i = 0 \rightarrow \text{length}(a)$ do
    $M(i, 0) \leftarrow d \cdot i$
for $j = 0 \rightarrow \text{length}(b)$ do
    $M(0, j) \leftarrow d \cdot j$
for $i = 1 \rightarrow \text{length}(a)$ do
    for $j = 1 \rightarrow \text{length}(b)$ do
        $v_1 \leftarrow M(i-1, j-1) + S(a_i, b_i)$
        $v_2 \leftarrow M(i, j-1) + d$
        $v_3 \leftarrow M(i-1, j) + d$
        $M(i, j) \leftarrow \max(v_1, v_2, v_3)$
```

two items.

Finally, the optimal alignment score of two sessions is normalized with the length of longer session [51]. The similarity score of two sessions S_i and S_j is defined as

$$sim(S_i, S_j) = \frac{\text{optimal alignment score of } S_i \text{ and } S_j}{\max(len(S_i), len(S_j))} \tag{3.18}$$

3.7 Clustering

To get usage profiles, sessions are clustered [8, 2, 5]. In this study, after defining the measure of similarity between two sessions, we apply clustering to get meaningful partitions of user sessions too.

The most widely used method for clustering is k-means algorithm. In k-means algorithm, clusters are represented with centroids. However, finding a centroid for sequence data is difficult. In [5], in order to find cluster centroids, objects in sessions are aggregated. Sequences are aligned and some objects are selected for centroid sequence at each step. A sample cluster containing 3 sequences is given in Table 3.4. The objects in step$_1$ have suport values of o_1: 66%, o_2: 100%, o_3: 33% and o_5: 33%. With support threshold 50%, objects selected for step$_1$ of centroid sequence are o_1 and o_3. For step$_2$ and step$_3$, by calculating support values and selecting ones with

Table 3.4: A sample cluster of sessions

	step$_1$	step$_2$	step$_3$
S_1	o_1, o_2	o_1, o_2, o_3	
S_2	o_2, o_3, o_5	o_4, o_5, o_8	o_1, o_2, o_3
S_3	o_1, o_2	o_1, o_3	

Table 3.5: Another sample cluster of sessions

	step$_1$	step$_2$	step$_3$
S_1	o_1, o_2	o_3, o_4	o_5, o_6
S_2	o_3, o_4	o_5, o_6	
S_3	o_7, o_8	o_1, o_2	o_3, o_4

support values greater than 50%, the centroid sequence for that cluster would be $\{o_1, o_2\} \rightarrow \{o_1, o_3\} \rightarrow \{o_1, o_2, o_3\}$.

In some cases, finding the cluster mean with this method is not easy. Consider the sample cluster of sessions in Table 3.5. They are very similar to each other. step$_1$ and step$_2$ of S_1 are same with step$_2$ and step$_3$ of S_3. step$_2$ and step$_3$ of S_1 are identical with step$_1$ and step$_2$ of S_2, However, with support threshold of 50%, the mean sequence does not contain any object for any step, because none of objects in any step has support greater that 50%.

3.7.1 CLUTO: a Clustering Tool

In this study, we used CLUTO clustering software [52] which is freely available[4]. There are two components of the software called *vcluster* and *scluster* which are used for clustering in vector space and similarity space, respectively. A clustering algorithm is applied on the set of objects with the aim of maximizing or minimizing a specific *clustering criterion function*. The clustering operation is treated as an optimization of the selected criterion function. Selected clustering method tries to optimize selected function to make the clustering solution better iteratively.

[4] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

There are several algorithms available in CLUTO package [53]. They are briefly given below. All these algorithms are greedy approaches and they make some criterion function locally optimum at each step.

- **Direct k-way clustering**: is similar to traditional k means clustering algorithm. Like k means, it iteratively refines the clustering until no change. The algorithm is given in Algorithm 4.

Algorithm 4 Direct k-way clustering algorithm that is implemented in CLUTO

```
(1) Randomly select $k$ objects as seed objects of $k$ clusters.
(2) Assign each object to the most similar seed objects.
(3) In a random order, assign each object to another cluster if it makes an improvement for the criterion function.
(4) Recalculate the centroid of each cluster using member data points of it.
(5) Repeat step 3 until no change in clusters.
```

- **Repeating bisections method** considers entire set of objects as one cluster at the beginning and obteions k clusters by repeating the operation of bisecting a cluster, $k-1$ times. At each step, one of the clusters is selection for bisection such that it makes the criterion function optimum.

- **Agglomerative method** starts with n clusters where n is the number of objects. At each step, two clusters are merged based on a particular criterion function. The algorithm runs until the number of clusters is k.

- **Graph-based method** models the set of objects using a nearest-neighbor graph [54]. In this graph, each object is represented with a vertex and objects are connected to their most similar objects with edges. This method splits the graph into k clusters using a min-cut graph partitioning algorithm.

In *vcluster* different similarity measures can be used for clustering. They are

- Cosine similarity,

- Eucledian distance,

Table 3.6: Definitions of CLUTO's clustering criterion functions. The notation in these functions is as follows: k is the total number of clusters, S is the total objects to be clustered, S_i is the set of objects assigned to the ith cluster, n_i is the number of objects in S_i, u and v represent two objects and $\text{sim}(u, v)$ is the similarity between objects.

Criterion Function	Optimization Function
$\mathcal{I}_1$	$\text{maximize} \sum_{i=1}^{k} n_i \left(\sum_{u,v \in S_i} \text{sim}(u, v) \right)$
$\mathcal{I}_2$	$\text{maximize} \sum_{i=1}^{k} \sqrt{\sum_{u,v \in S_i} \text{sim}(u, v)}$
$\mathcal{E}_1$	$\text{minimize} \sum_{i=1}^{k} n_i \frac{\sum_{u \in S_i, v \in S} \text{sim}(u, v)}{\sqrt{\sum_{u,v \in S_i} \text{sim}(u, v)}}$
$\mathcal{G}_1$	$\text{minimize} \sum_{i=1}^{k} \frac{\text{cut}(S_i, S - S_i)}{\sum_{u,v \in S_i} \text{sim}(u, v)}$
$\mathcal{H}_1$	$\text{maximize} \frac{\mathcal{I}_1}{\mathcal{E}_1}$
$\mathcal{H}_2$	$\text{maximize} \frac{\mathcal{I}_2}{\mathcal{E}_1}$

- Correlation coefficient and

- Jaccard coefficient.

Their definitions are given in Section 2.7.2. Unlike *vcluster*, *scluster* does not need selection of a similarity measure since it takes input of similarity matrix between objects to be clustered.

Several criterion functions are available to use in CLUTO. For measuring the quality of clusters, they take separation between clusters and tightness of each cluster into account. They are given in Table 3.6.

scluster component of CLUTO does not require cluster means, it uses the similarity matrix of sequences. We used criterion function $\mathcal{H}_2$ for clustering with repeated bisections clustering algorithm. The reason of choosing these settings is that $\mathcal{H}_2$ measures both intra-cluster and inter-cluster similarity. Also, previously, these clustering algorithms were compared with different criterion functions in [53, 55]. They studied these criterion functions for clustering document datasets and they found that $\mathcal{H}_2$ achieved the best overall results with the repeated bisections algorithm.

CHAPTER 4

EXPERIMENTS

The dataset used in this study is the server logs of the web site `http://www.ceng.metu.edu.tr`, the web site of Middle East Technical University, Department of Computer Engineering. All accesses requested to the server are between 06 February 2011 and 18 February 2011. They were collected in Apache HTTP server combined log format. The total size of logs is 107 MB.

Preprocessing steps explained in Section 3.1 were followed and sessions were identified for clustering. Some access logs were removed from dataset during preprocessing because of one of the followings reasons.

- The log item is either multimedia file, archive file or an external document which is not useful for further processing.
- The log item belongs to a web crawler.

Initially there are 293969 access log items. After removal of logs of specific file types and removal of logs belonging to web crawlers, the number of log items is 33690. At the end of preprocessing, the number of unique IP addresses is 3538.

After removal of irrelevant log items, 4371 unique URLs were fetched from the server and they were mapped to 301 concepts in the concept taxonomy. The average number of concepts assigned to a URL is 2.87±4. The maximum number of concepts associated with a URL is 45 and the minimum number of concepts assigned to a URL is 1. The reason for high standard deviation ($\sigma = 4.0$) is the pages that contain kind of general information. For example, home page contains more information than an average page contains, the web page of list of faculty members contains research interests of everyone as well, so it contains several keywords which are mapped to several concepts.

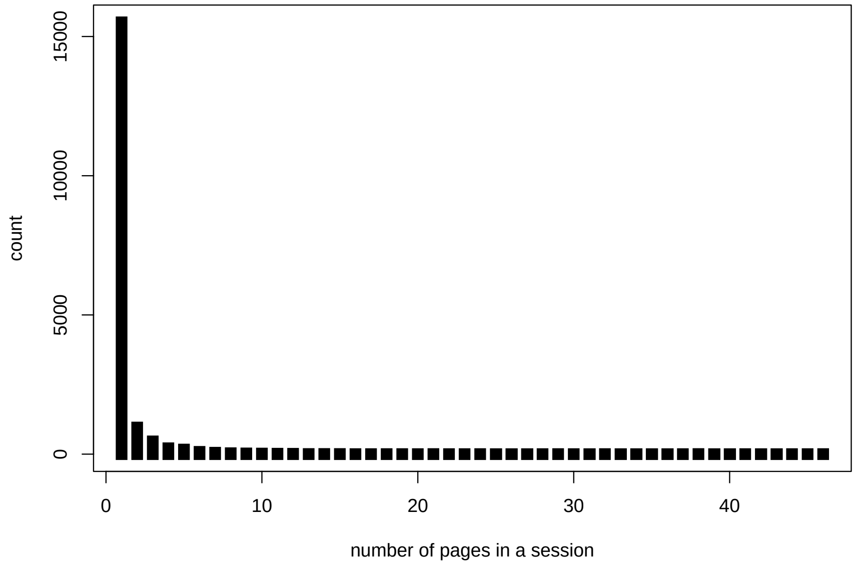
The next step is session construction. Using IP address and time fields of log items, sessions were constructed. Statistics of sessions are given in Table 4.1. The total number of sessions is 19063. Average number of page views in a session is 1.97 ± 8.67. Average length of a session is 5.72 ± 20.21 seconds. In addition to these statistics, the histogram of number of pages in a session and the histogram of session length (in seconds) are given in Figure 4.1 and Figure 4.2 respectively. For better comparison of frequencies of number of pages, histograms in log-scale are also given in these figures. Figure 4.1 shows that sessions usually contain small number of pages, less than 5 in most cases. In Figure 4.2, it is seen that most sessions last less than a few seconds, like most sessions consist of a few pages. It can be concluded that a user do not usually spend so much time on the web site and getting useful sessions to use for getting useful patterns is not easy.

Table 4.1: Some statistics of the dataset

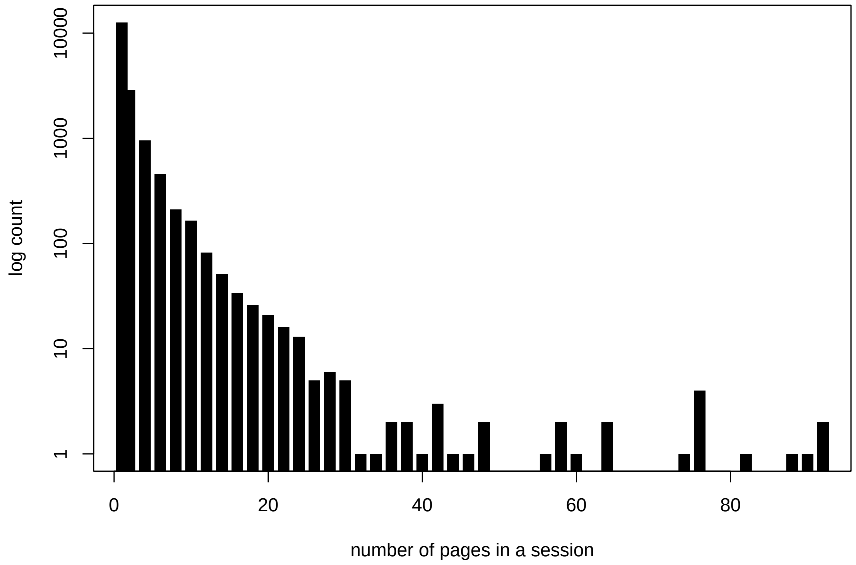
number of sessions	19063
number of page views in a session	1.97 ± 8.67
length of a session [in seconds]	5.72 ± 20.21

For the measurement of similarity between two concepts, Rada et al.'s distance was used [40]. Rada et al.'s distance between two concepts is the length of the shortest path connecting two concepts in the concept taxonomy. Therefore, the similarity between two concepts is $1/d$, where d is the Rada et al.'s distance between two concepts (see 3.4). In this study, Rada et al.'s distance was selected because it is simple to apply and it is well suited to measure similarity among a set of concepts represented with a tree.

For the similarity between two web pages, average linkage based similarity was used (see Section 3.5). Concepts that are mapped to web pages were used. Let P_a be defined with concept set C_a, P_b be defined with concept set C_b. The similarity between P_a and P_b is the average of similarities of all pairs of concepts c_a and c_b, where $c_a \in C_a$

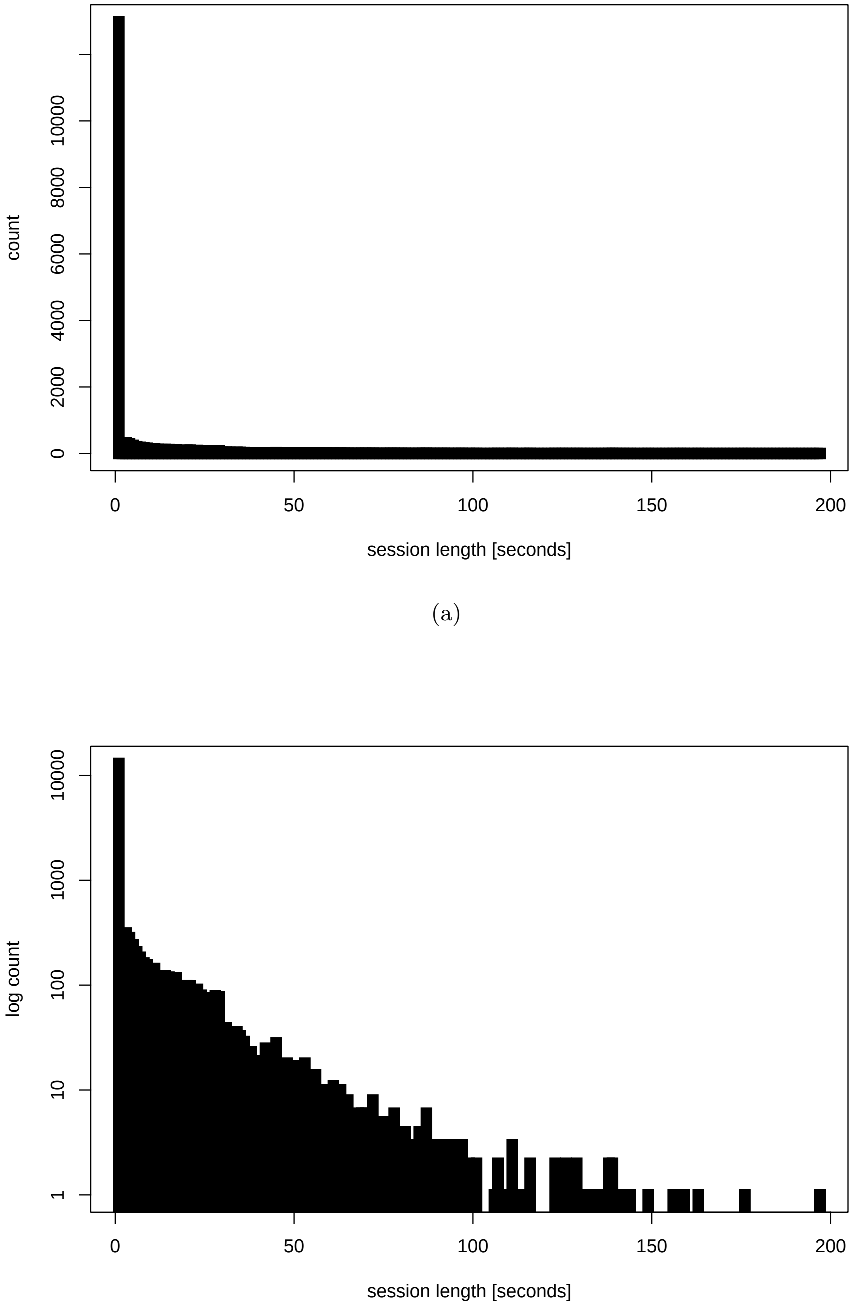


(a)



(b)

Figure 4.1: Histogram of number of pages in sessions is given in figure (a), histogram of number of pages in sessions in log scale is given in figure (b).



(a)

(b)

Figure 4.2: The histogram of session lengths in seconds. For easy comparison, histogram of session lengths in log scale is given in figure (b).

and $c_b \in C_b$. In this study, we use average linkage based similarity, because it takes all similarities of pairs of concepts into account. Also, it is the most common one used for set similarity measurement.

Next step is to calculate similarities between two sessions. Since we incorporate the sequence information (i.e. the order of visited web pages), sessions with length less than 4 were removed. The total number of sessions with length at least 4 is 1126.

As explained in Section 3.6, to measure the similarity between two sessions, Needleman-Wunsch alignment algorithm was used [50]. Similar to [5], the algorithm was run with gap penalty 0.

For implementation, Python was used as programming language. Experiments were run on a 2.30 GHz machine with 8 GB memory. The preprocessing step took 12 seconds. One of the longest steps is getting contents of web pages. It highly depends on connection speed. With a computer in the same network of the web site server, it took 19 minutes to download 4371 unique URLs. The computational complexity of calculating all pairwise concept similarities (i.e. using Rada et al.'s distance for each pairwise concepts) is $\mathcal{O}(dn^2)$ where d is the depth of concept hierarchy tree and n is the number of all concepts used to describe web pages. It took almost a minute. The next step is to calculate pairwise session similarities. For each session, the similarity to all other sessions were calculated. For each similarity calculation, Needleman-Wunsch dynamic programming was run between two sessions (i.e. ordered set of pageviews where each pageview is a set of concepts itself). After getting all measurements between sessions, running CLUTO did take less than a few seconds.

Further analysis on the dataset were done in two parts. For the first part of the experiments, the temporal information in sequences is not used. In other words, only the similarity component $\mathcal{S}'$ is used (see Section 3.5). After first set of experiments, the importance component which uses time spent on each page view was introduced. Results with and without the importance component are compared later to see the effect of using duration information on clustering clickstream sequences.

For clustering, CLUTO clustering toolkit was used [52]. CLUTO has several different clustering algorithms with several different criterion functions (see Section 3.7.1). In this study, repeated bisections was selected as clustering algorithm with $\mathcal{H}_2$ criterion function (see Section 3.7.1). The reason for choosing these settings is that repeated bisections gives better results with $\mathcal{H}_2$ criterion function in the domain of document datasets [53, 55].

Since determining k, the optimal number of clusters is not easy, CLUTO was run with different values from $k = 2$ to $k = 50$. For each k, the inter-cluster similarity and intra-cluster similarity values were calculated.

The best clustering is the one with maximum intra-cluster similarity and minimum inter-cluster similarity. In this study, to evaluate clusterings we use the difference of intra-similarity score $IS(\mathcal{C})$ and inter-similarity score $ES(\mathcal{C})$ values. The evaluation score for a clustering $\mathcal{C}$, $\mathcal{S}(\mathcal{C})$ is

$$\mathcal{S}(\mathcal{C}) = IS(\mathcal{C}) - ES(\mathcal{C}) \tag{4.1}$$

In following two subsections, experiment results with and without temporal component (i.e. importance component) are given.

4.1 Experiments Using Only Semantic Similarity

Clustering evaluation score for different k values are given in Figure 4.3. The clustering with the highest quality is when $k = 20$. For the rest of experiments, the clustering at $k = 20$ is used.

The number of sessions in each cluster is given in Figure 4.4. The average number of sessions is 33.4 ± 20.7. Clusters in Figure 4.4 are ordered in decreasing $IS(C) - ES(C)$ order, where $IS(C)$ is the average similarity between the objects of cluster C and $ES(C)$ is the average similarity of objects of the cluster C and the rest of the objects outside C. Therefore, the cluster #1 is the most tight and the most far away from the rest of the objects in clusters with large cluster ids. The clusters with small ids are smaller (i.e. contain less number of objects) than clusters with larger ids.

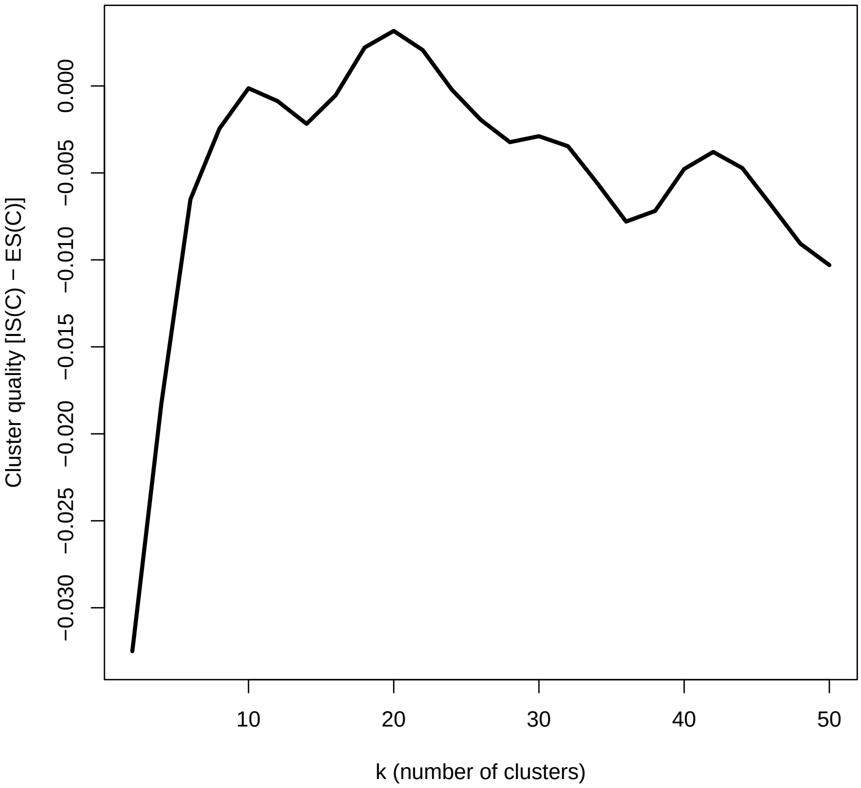


Figure 4.3: Cluster evaluation scores $(IS(\mathcal{C}) - ES(\mathcal{C}))$ for different k values. The maximum score is when $k = 20$.

The Figure 4.5 shows the average number of page views in a session for each cluster. R statistical computing software [56] was used for box-and-whisker plots which are also known as box plots. After removal of outliers, the horizontal line in the box shows the median of the data. The bottom edge and top edge of the box represent the first and third quartiles respectively. The first quartile (Q_1) of the data is the element for which 25% of the data is less than it. Similarly the third quartile (Q_3) is the element for which 75% of the data is less than that element. The horizontal lines above and below the box represent maximum and minimum respectively. In Figure 4.5, lengths of sessions in the same cluster are usually small and close to each other. However, three clusters, #4, #10 and #16 have high variance of session lengths (in number of page views). Sizes of these clusters are relatively small when compared with other ones, especially clusters with small variances (see Figure 4.4).

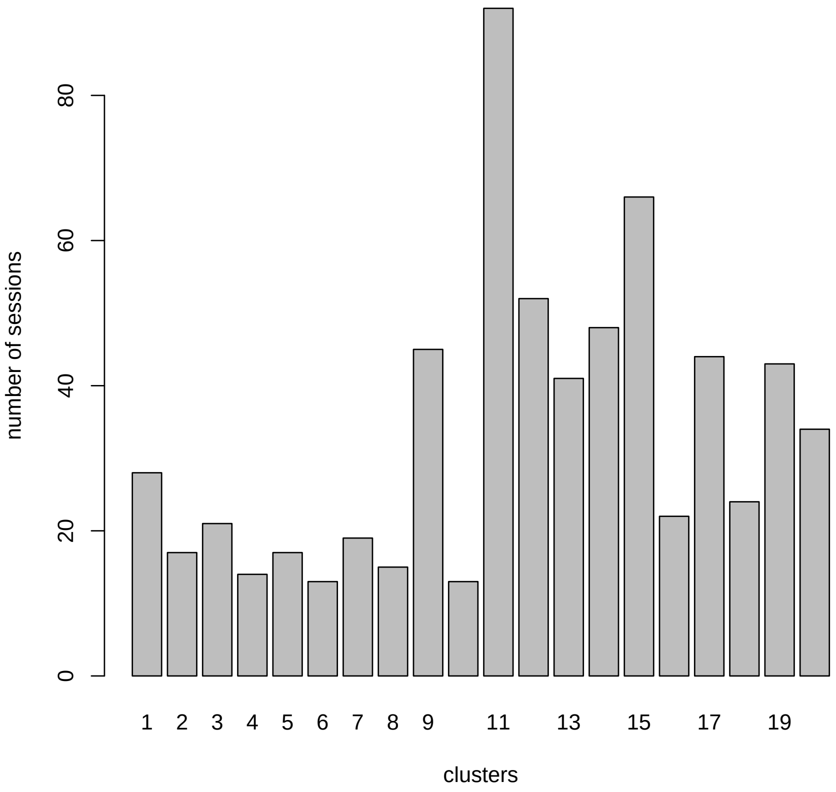


Figure 4.4: The number of sessions in each cluster (without time-spent information)

The box plot in Figure 4.6 shows the average length of sessions (in seconds) for each cluster. As expected, there is positive correlation between session lengths in number of page views and session lengths in seconds. Clusters #10 and #16 have higher average session lengths in seconds like they have higher average session lengths in number of page views (see Figure 4.5). Although its average session length in number of page views is larger, cluster #4 has not so large average session length in seconds. The reason is that it contains sessions with long number of page views but duration of these page views are small.

Table 4.2 shows most frequent concepts seen in each cluster. As expected, some of them are general concepts which are close to the root of the concept taxonomy. Concepts are given together with their depths in concept taxonomy tree. In addition

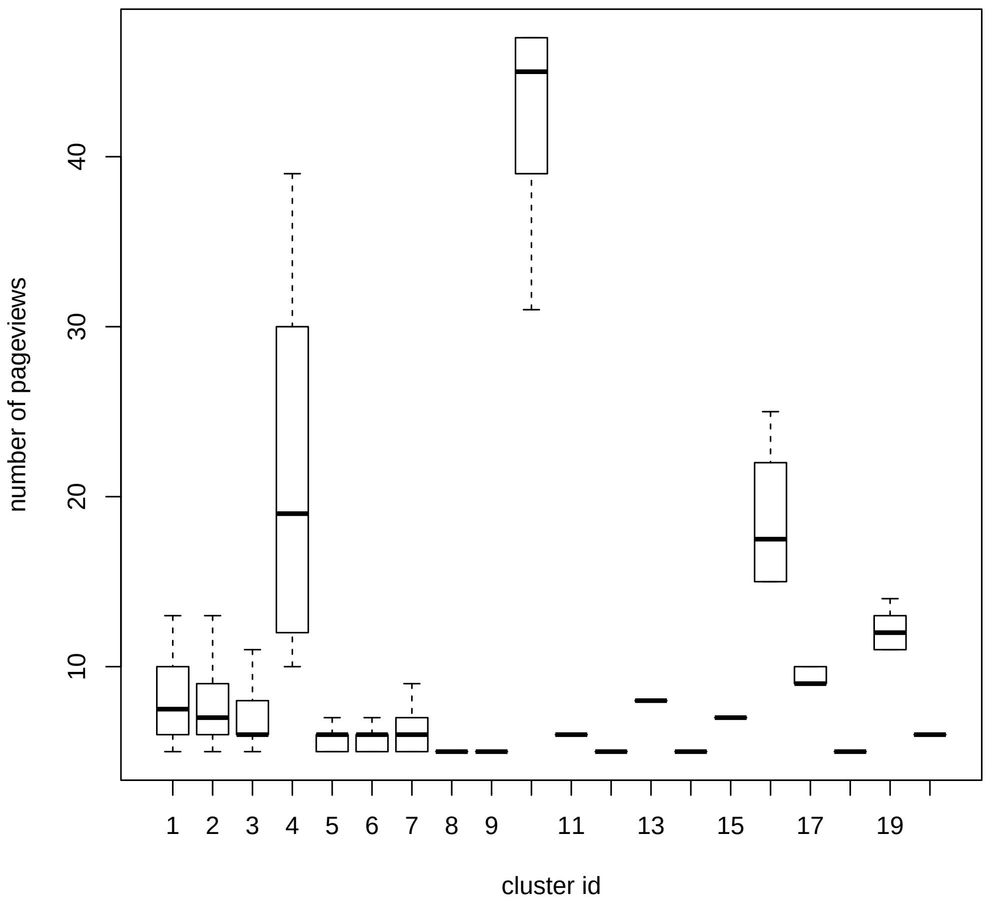


Figure 4.5: The average number of page views in a session for each cluster

to that more specific concepts (with depth 3 or more) are given in bold.

4.2 Experiments Using Time-spent Information with Semantic Similarity

In the second part of experiments, time-spent information (importance component) is introduced to the session similarity measure (see Section 3.5). To compare results with previous experiment results (ones without using importance component), the number

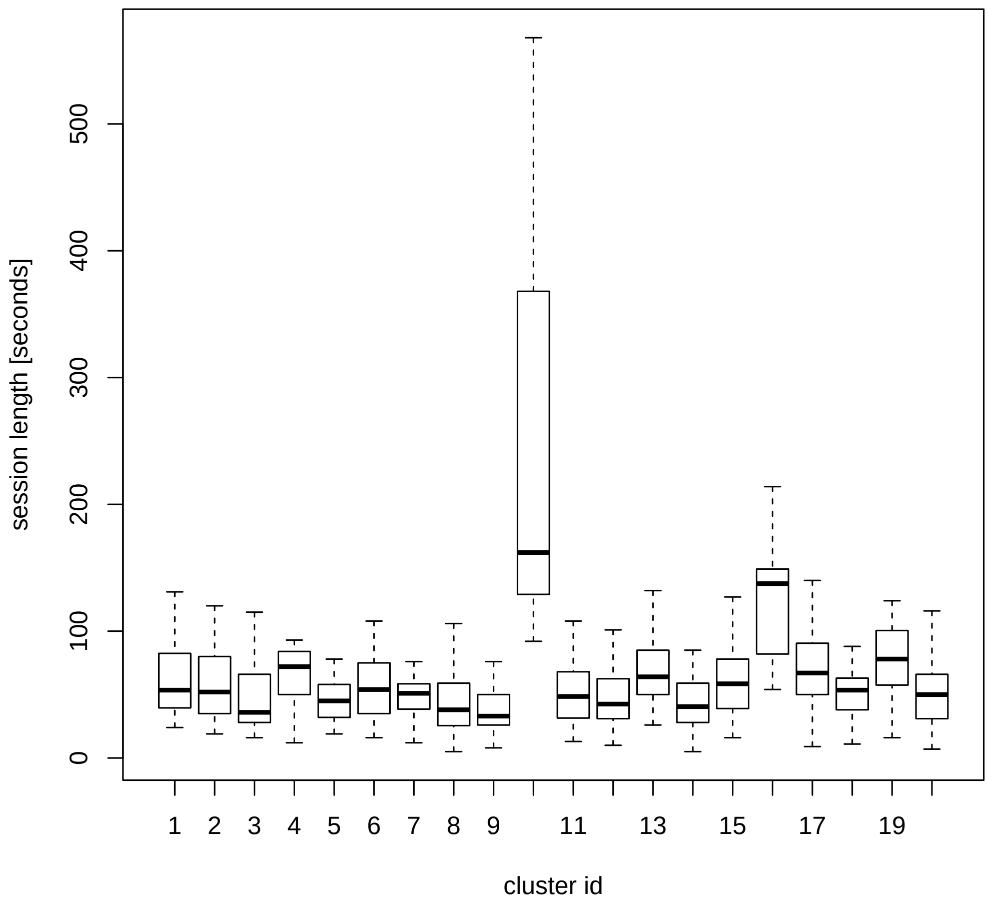


Figure 4.6: The average session length for each cluster (without time-spent information)

of clusters (k) is set to 20.

Figure 4.7 shows the number of sessions in each cluster. The average number of sessions in a cluster is again 33.4, since total number of sessions and k are same with experiments that are run without using time-spent information. However, the standard deviation is 7.52, much less than first experiment (which is 20.7). It shows that the clustering computed using time-spent information is more balanced, that is items are assigned to clusters more evenly.

Table 4.2: Most frequent concepts for each cluster (without time-spent information)

Cluster	Most frequent concepts
1	**akyuz**(6), Algorithms(2), Graphics(2), **ComputerGraphics**(4), PL(2), Research(2), **ceng242**(4)
2	Course(2), **ceng140**(4), PL(2), Research(2), Graphics(2), **ComputerGraphics**(4), Thesis(2)
3	PL(2), **ceng242**(4), Algorithms(2), Course(2), TechnicalReport(2), Book(2), **ceng140**(4), DataStructures(2), Research(2)
4	**gokdeniz**(5), Research(2), Schedule(1), Bioinformatics(2), PatternRecognition(2), AI(2), **Seminar**(3)
5	**ucoluk**(6), Research(2), AI(2), PL(2), **Seminar**(3), **Kovan**(4), Schedule(1), SoftwareEng(2), Graphics(2)
6	Research(2), AI(2), **Seminar**(3), **erdal**(5), Schedule(1), Bioinformatics(2), PatternRecognition(2), **NewsArchive**(3)
7	Research(2), SoftwareEng(2), **Seminar**(3), AI(2), **Doctor**(4), **volkan**(6), Thesis(2), Bioinformatics(2)
8	Course(2), Research(2), PL(2), Book(2), AI(2), **ResearchLaboratory**(3), **Kovan**(4), Schedule(1), SoftwareEng(2)
9	Schedule(1), Research(2), **NewsArchive**(3), **Seminar**(3), **Magazine**(3) Course(2), **GraduateCourse**(3), Thesis(2)
10	Schedule(1), PatternRecognition(2), Research(2), SoftwareEng(2), Course(2), Bioinformatics(2), Graphics(2), AI(2), **Seminar**(3), **ResearchLaboratory**(3)
11	Research(2), Schedule(1), **Seminar**(3), **NewsArchive**(3), **Magazine**(3), AI(2), PatternRecognition(2), Graphics(2), **ComputerGraphics**(4)
12	Research(2), **NewsArchive**(3), Schedule(1), **Seminar**(3), **Magazine**(3), AI(2), PatternRecognition(2)
13	Research(2), AI(2), **Seminar**(3), Schedule(1), **NewsArchive**(3), PatternRecognition(2), Course(2), SoftwareEng(2)
14	Research(2), **Seminar**(3), AI(2), Schedule(1), PatternRecognition(2), **NewsArchive**(3), SoftwareEng(2)
15	Research(2), **Seminar**(3), Schedule(1), AI(2), **NewsArchive**(3), **Magazine**(3), **Modsim**(4), Course(2), Thesis(2)
16	Research(2), Schedule(1), AI(2), **Seminar**(3), PatternRecognition(2), **ResearchLaboratory**(3), **NewsArchive**(3), **Magazine**(3)
17	Research(2), AI(2), **Seminar**(3), **NewsArchive**(3), Schedule(1), SoftwareEng(2), PatternRecognition(2), ParellelComputation(2), Student(2)
18	Research(2), AI(2), SoftwareEng(2), **Doctor**(4), Algorithms(2), NLP(2), **Kovan**(4), Thesis(2), Bioinformatics(2), **ResearchLaboratory**(3)
19	Research(2), AI(2), **Seminar**(3), Schedule(1), **NewsArchive**(3), Course(2), PatternRecognition(2), **Magazine**(3)
20	Research(2), SoftwareEng(2), AI(2), PatternRecognition(2), Course(2), Thesis(2), Bioinformatics(2), Schedule(1)

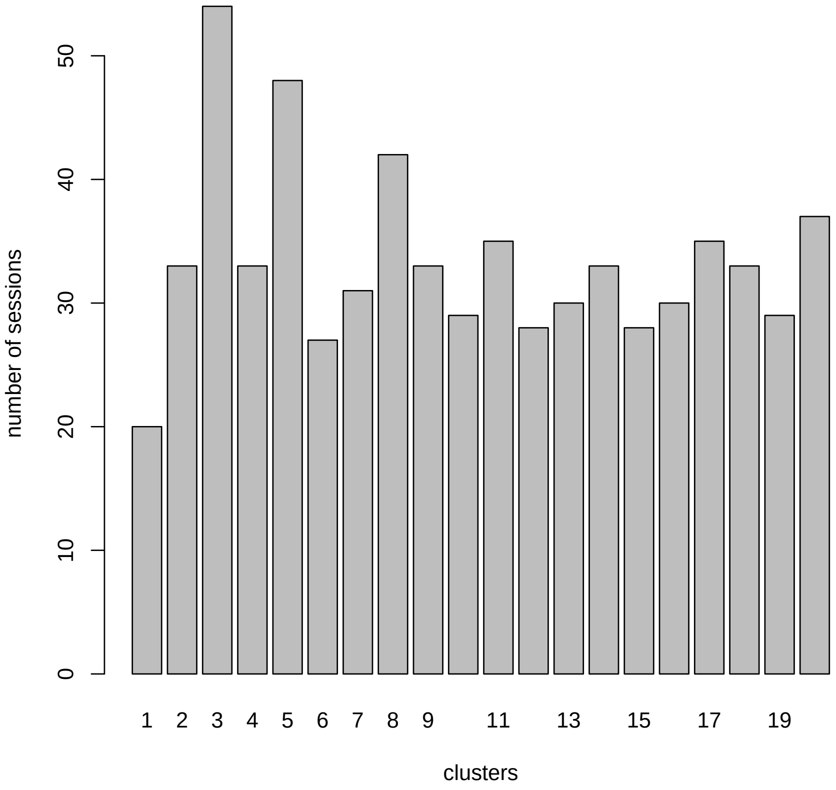


Figure 4.7: The number of sessions in each cluster (with time-spent information)

Figure 4.8 shows the average session length in number of page views for each cluster. For most of the clusters, variance of number of page views in sessions in the same cluster is low. Cluster #17 has a high median number of page views as well as high variance. Similar to the first set of experiments, clusters in figures are ordered in decreasing $IS(C) - ES(C)$ order, where $IS(C)$ and $ES(C)$ are the internal and external similarity of the cluster C, respectively. Therefore, cluster #17 which has high median and variance is computed with relatively low confidence.

Figure 4.9 shows the average session length too, but in seconds. Median of session lengths are more close to each other when compared with Figure 4.6, the result of experiment without using time-spent information component. Like page view lengths, cluster #17 has the highest median, which shows the correlation betwen session length

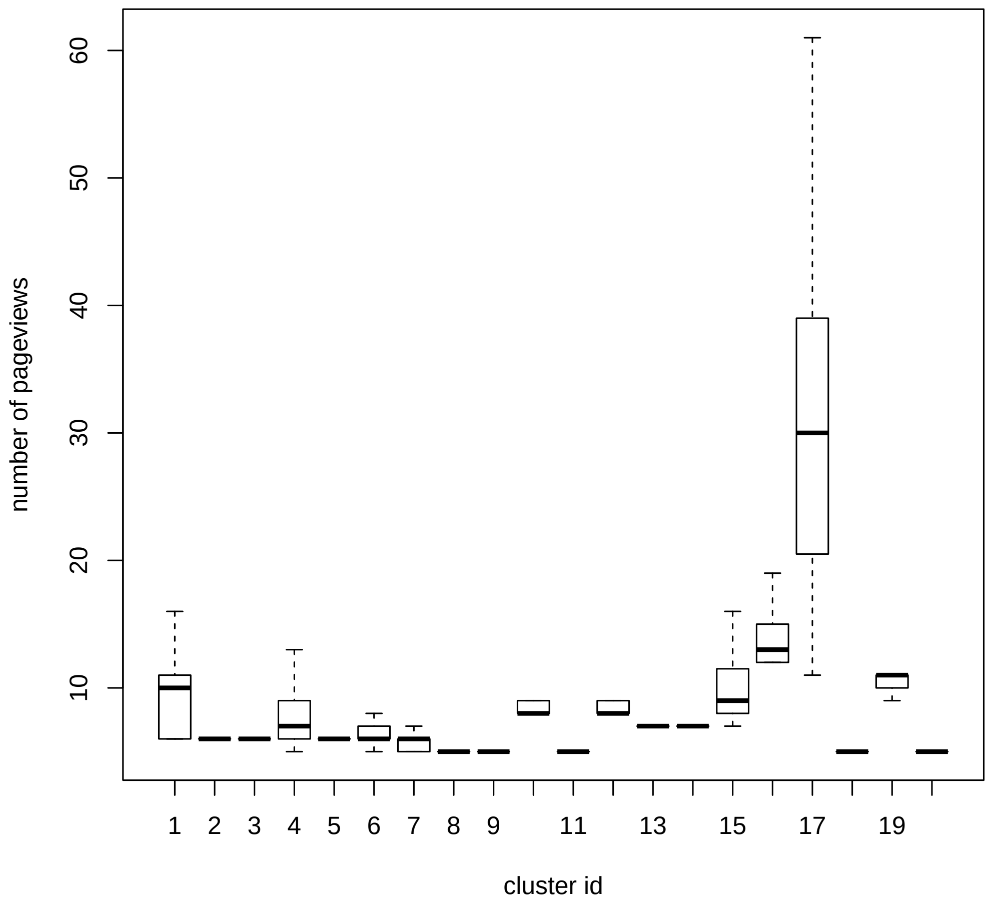


Figure 4.8: The average number of page views in a session for each cluster (with time-spent information)

in number of page views and the session length in seconds.

Table 4.3 shows the most frequent concepts for each cluster. Similar to the Table 4.2, most of concepts are high level concepts in the concept taxonomy. Like Table 4.2, concepts are given with their depth in concept taxonomy. Also, more specific concepts (with depth 3 or more) are given in bold.

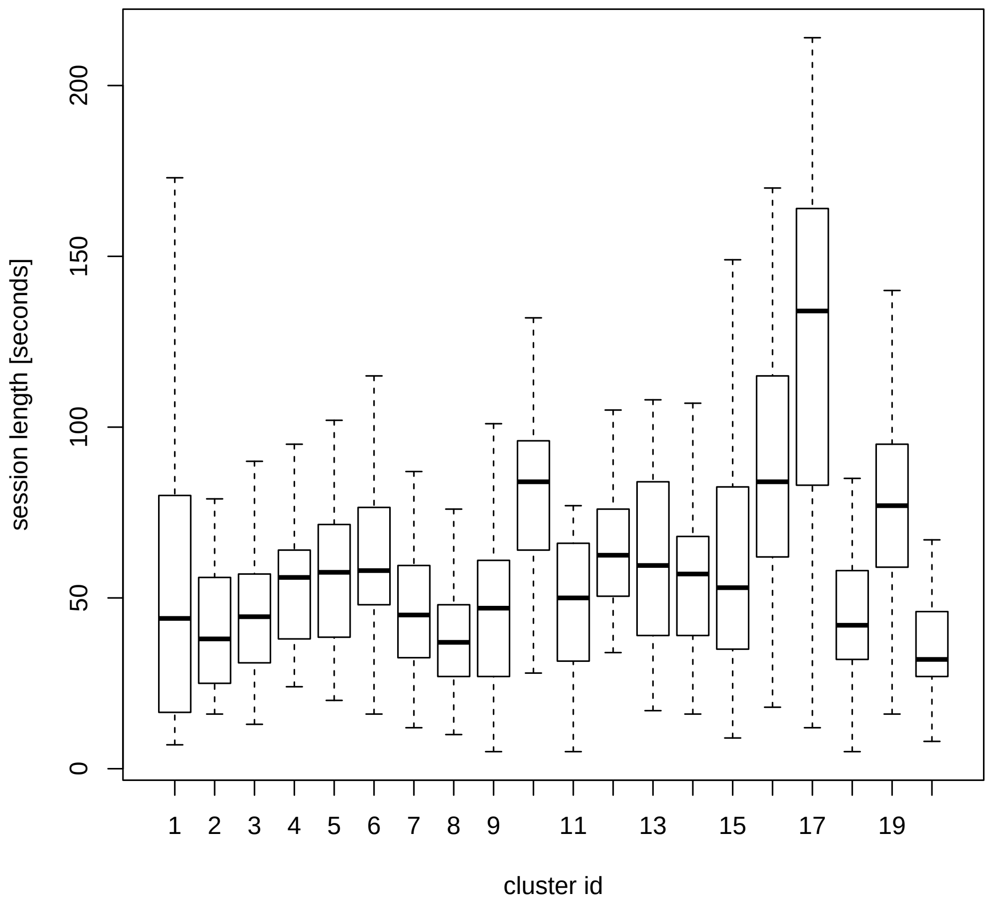


Figure 4.9: The average session length for each cluster (with time-spent information)

4.3 Recommendation Experiments

The clustering statistics and most frequent concepts given in previous sections of this chapter are helpful to understand data. However, it is difficult to compare experiment results by looking those experiment results. To evaluate the method used in this study, accuracy of recommendation of new web pages is used. To evaluate the similarity measure used to compute similarity between two sessions, recommendation (prediction) experiment was performed. Let $S = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C} \rightarrow \texttt{D}$ be a session of length 4. For recommendation experiments the last item of the session (in this case page D) is removed from the session and it is tried to be predicted by using other

Table 4.3: Most frequent concepts for each cluster (with time-spent information)

Cluster	Most frequent concepts
1	Research(2), AI(2), SoftwareEng(2), PatternRecognition(2), Thesis(2), **Seminar**(3), Schedule(1), ParellelComputation(2)
2	Research(2), Schedule(1), **Seminar**(3), **NewsArchive**(3), **Magazine**(3), AI(2), **Modsim**(4), PatternRecognition(2)
3	Course(2), **ceng242**(4), PL(2), **ceng140**(4), Research(2), NLP(2), Algorithms(2), Schedule(1)
4	**akyuz**(6), Algorithms(2), Graphics(2), **ComputerGraphics**(4), PL(2), Research(2), Course(2), SoftwareEng(2)
5	Schedule(1), Research(2), **Seminar**(3), **NewsArchive**(3), **Magazine**(3), AI(2), Graphics(2), **ComputerGraphics**(4)
6	Research(2), Bioinformatics(2), AI(2),**Seminar**(3), Schedule(1), Thesis(2), **Modsim**(4), Graphics(2)
7	Research(2), AI(2), SoftwareEng(2), Thesis(2), PatternRecognition(2), Algorithms(2), Graphics(2), **Seminar**(3)
8	Research(2), AI(2), Schedule(1), **Seminar**(3), **NewsArchive**(3), **Magazine**(3), PatternRecognition(2), NLP(2)
9	Research(2), Course(2), **Seminar**(3), **NewsArchive**(3), Schedule(1), AI(2), **GraduateCourse**(3), SoftwareEng(2)
10	Schedule(1), Research(2), **Seminar**(3), **NewsArchive**(3), **Magazine**(3), **Modsim**(4), AI(2), PatternRecognition(2)
11	Schedule(1), Research(2), **NewsArchive**(3), **Seminar**(3), **Magazine**(3), AI(2), PatternRecognition(2), NLP(2)
12	Research(2), AI(2), SoftwareEng(2), PatternRecognition(2), **Kovan**(4), **ComputerGraphics**(4), ParellelComputation(2), Graphics(2)
13	Research(2), Schedule(1), **Seminar**(3), **NewsArchive**(3), AI(2), Course(2), **GraduateCourse**(3), **erdal**(5)
14	Research(2), PatternRecognition(2), AI(2), NLP(2), **Seminar**(3), SoftwareEng(2), Bioinformatics(2), Graphics(2)
15	Course(2), Student(2), Research(2), AI(2), SoftwareEng(2), PL(2), Graphics(2), **ComputerGraphics**(4)
16	Research(2), **Kovan**(4), AI(2), **ResearchLaboratory**(3), Schedule(1), **Seminar**(3), **NewsArchive**(3), **Magazine**(3)
17	Schedule(1), Research(2), PatternRecognition(2), AI(2), **Seminar**(3), **ResearchLaboratory**(3), **Kovan**(4), NLP(2)
18	Research(2), Course(2), SoftwareEng(2), AI(2), **Kovan**(4), **ResearchLaboratory**(3), Graphics(2), **ComputerGraphics**(4)
19	Schedule(1), Research(2), **Seminar**(3), **NewsArchive**(3), SoftwareEng(2), Thesis(2), AI(2), Course(2)
20	Research(2), AI(2), Graphics(2), **Seminar**(3), **ComputerGraphics**(4), Schedule(1), SoftwareEng(2), Bioinformatics(2)

sessions in the dataset. To compare session $S' = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$ with other sessions in the dataset, the similarity measures with/without time information are used. At the end, if page `D` is in the set of recommendations, the recommendation is considered as accurate, otherwise not accurate.

For recommendation experiment, 5-fold cross validation was used. The dataset (the set of sessions of length at least 4) was divided into 5 equal partitions. Each turn, one of these partitions is used for testing and the rest (four-fifth) of the dataset is used for training. The experiment was repeated 5 times and different one of the partitions was used for testing each time. At the end, each session in the dataset was used for testing only once.

For recommendation, k-nearest neighbor algorithm was used. Given a test session S, the last URL in S is removed for prediction. k sessions from training set that are most close to S (without last URL) are selected and the last URL of S is tried to be predicted based on these k nearest neighbor sessions.

Example Let $S = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C} \rightarrow \texttt{D}$ be the test session. The last item of it is removed from the session to be used for recommendation, so $S' = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$ is mapped to concept set sequence and k most-similar sessions are selected from the training set. Let $k = 2$ and the closest sequences be $S_1 = \texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{E} \rightarrow \texttt{F}$ and $S_2 = \texttt{A} \rightarrow \texttt{C} \rightarrow \texttt{D}$. S' is aligned with S_1 and S_2 separately. The alignment of S' and S_1 is

$$\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$$

$$\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{E} \rightarrow \texttt{F}$$

The next web page of S' is predicted as page F. The alignment of S' and S_2 is

$$\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C}$$

$$\texttt{A} \rightarrow \texttt{-} \rightarrow \texttt{C} \rightarrow \texttt{D}$$

The next web page of S' is predicted as page D. Therefore, the set of predictions as the next web page pf S' is $\{\texttt{F}, \texttt{D}\}$. Since the real next web page is `D` is in prediction set, the recommendation is considered as accurate.

The session similarity measure used in this study was compared with similarity measure looking URLs to assess similarity. According to this measure, the similarity between two web pages is 1.0 if their URLs are same, 0.0 otherwise. This similarity measure (called URL-equality measure from this point) was compared with our similarity measures which use URL mapping to concept tree and Rada et al.'s distance with and without time-spent information.

For different k values recommendation accuracies are given in Figure 4.10. Recommendation accuracy is the ratio of correct predictions to the sum of correct and false predictions. exp1 is results by using URL-equality similarity measure. For smaller k values, the recommendation accuracy is very low and around 0.1. For larger k values, the recommendation accuracy increases to 0.2, since the number of recommendations is k and it is more likely to predict next web page true with more predictions. The line exp2 shows results of experiments by using our similarity measure *without* using time-spent information. It is much more accurate than URL-equality similarity. For $k = 10$, the prediction accuracy is around 65%. The last line, exp3 shows results of experiments using our similarity measure *with* using time-spent information. The results are very close to results of similarity measure without time-spent information. For $k = 10$, similarity with time-spent gives better results.

To find out whether performance differences of similarity measures are significant or not, t-test was performed. $k = 10$ is selected for t-test. Each experiment was run 50 times. First, URL-equality measure was compared with our similarity measure *without* time-spent information. $p = 2.49E^{-79}$ suggests that means of recommendation accuracies of these two similarity measures are significantly different. Second, the effect of using time-spent information was analyzed. Our similarity measures *with* and *without* using time-spent information were compared and p-value was 0.525 which shows that they are *not* significantly different. Means and standard deviations of prediction accuracies are given in Table 4.4. t-test results show that the experiments of our similarity measure have more accuracy than URL-equality measure. However, using time-spent information does not increase or decrease performance significantly.

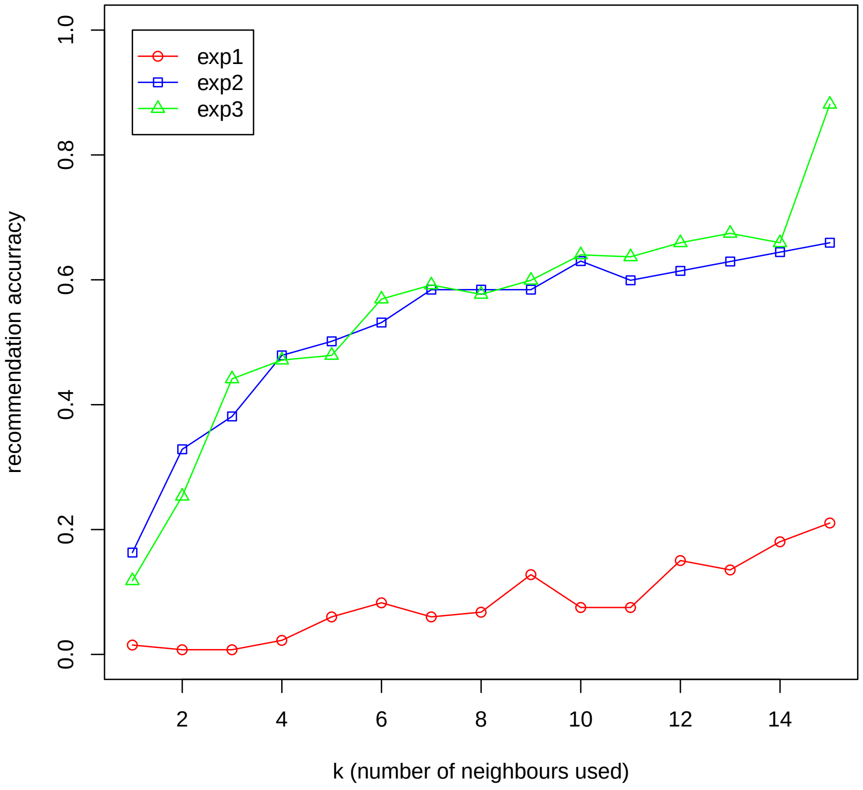


Figure 4.10: Prediction accuracy rates for different similarity measures. `exp1` is the URL-equality similarity measure. `exp2` and `exp3` are results of our similarity measures *without* and *with* using time-spent information, respectively.

Table 4.4: Means and standard deviations of prediction accuracies of three different similarity measures

similarity measure	mean	stdev
URL-equality	0.09564	0.0243
without time-spent info	0.6347	0.04002
with time-spent info	0.64003	0.042

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this study, session clustering and recommendation systems are presented. Using raw dataset of web page access logs, sessions of users are constructed and each web page on these sessions are mapped to set of concepts. Two different kinds of similarity measures are used: Rada et al.'s distance to measure the similarity between two concepts, and average set similarity to measure the similarity between two sets of concepts. Using these similarity measures, Needleman-Wunsch dynamic programming algorithm is used to measure similarity between two sessions, which is used for clustering of sessions.

Computed clusters represent groups of sessions of users with similar contents. These clusters can be used to understand the behavior of users. They can be used for recommendation of new pages to users during the web site navigation.

In Section 3.4, several semantic similarity measures are given. There are studies in the literature evaluating semantic similarity measures [39]. Most of them use the Gene Ontology[5] which describes genes and their products, or WordNet[6] which is a large lexical database of English. Applying different semantic similarity measures to web usage mining domain and experimental analysis and comparison of these measures is the future work of this study.

Another future work is the comparison of set similarity measures in web usage mining domain. In this study, usually web pages are represented with a set of concepts, not only one concept. Different measures for similarity of concept sets are given in Section

[5] http://www.geneontology.org

[6] http://wordnet.princeton.edu

3.5. In this study, only one of them is used, which is average set similarity. Experiment analysis and comparison of different set similarity measures is one another future work of this study.

The introduction of time-spent information to web page similarity measure is analyzed. The component that uses time-spent is based on the proportion of time spent at the web page to total session time. In other words, if measuring the similarity between pages P_A and P_B where $P_A \in S_A$ and $P_B \in S_B$,

- if time-spent on P_A is not significant in total time of session S_A, or
- if time-spent on P_B is very small when compared the time spent in whole session S_B,

the contribution of similarity (P_A,P_B) to similarity of sessions S_A and S_B is decreased. The exact definition of time-spent component is given in (3.7). This aspect of time-spent information does not look for the similarity of the amount of time spent between P_A and P_B. Another time component can be introduced such that the contribution of the similarity between P_A and P_B should be amplified if the time spent on P_A and P_B are close to each other.

Instead of using one clustering method, by choosing different combinations of semantic similarity measure, set similarity measure and clustering algorithm, different clusterings can be computed. To improve the quality of these clusterings, different partitionings of sessions by different experiment setups can be combined [57].

Another improvement is possible on preprocessing step. Better preprocessing heuristics can be selected to improve data quality which improves clustering and recommendation of the system. Path completion step can be introduced to make paths of web pages complete which are not due to caching mechanisms implemented in browsers or proxy servers.

In this study, before mapping of each web page to a set of concepts, some keywords are used to describe each concept. These keywords are searched in the web page and if necessary keywords are in the web page, the page is labeled with that concept. Keywords for each concept are selected manually. Another possible future work is to select these keywords automatically. In this study, 301 concepts are used. By selecting keywords and creating ontology automatically, much more concepts can be used to describe the domain with more specifically.

REFERENCES

[1] Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, and Constantine D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction*, 13:311–372, November 2003.

[2] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43:142–151, August 2000.

[3] Federico Michele Facca and Pier Luca Lanzi. Mining interesting knowledge from weblogs: a survey. *Data Knowl. Eng.*, 53:225–241, June 2005.

[4] Myra Spiliopoulou. Web usage mining for web site evaluation. *Commun. ACM*, 43:127–134, August 2000.

[5] Hakan Yilmaz and Pinar Senkul. Using Ontology and Sequence Information for Extracting Behavior Patterns from Web Navigation Logs. In *IEEE, ICDM Workshop on Semantic Aspects in Data Mining (SADM'10)*, Dec 2010.

[6] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the world wide web. In *ICTAI '97: Proceedings of the 9th International Conference on Tools with Artificial Intelligence*, page 558, Washington, DC, USA, 1997. IEEE Computer Society.

[7] Bamshad Mobasher, Namit Jain, Eui-Hong S. Han, and Jaideep Srivastava. Web Mining: Pattern Discovery from World Wide Web Transactions, 1996.

[8] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explor. Newsl.*, 1:12–23, January 2000.

[9] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. Discovery of interesting usage patterns from web data. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, WEBKDD '99, pages 163–182, London, UK, 2000. Springer-Verlag.

[10] Lita van Wel and Lamber Royakkers. Ethical issues in web data mining. *Ethics and Information Technology*, 6:129–140, 2004. 10.1023/B:ETIN.0000047476.05912.3d.

[11] Raymond Kosala and Hendrik Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2(1):1–15, 2000.

[12] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] M. G. da Costa and Zhiguo Gong. Web structure mining: an introduction. In *2005 IEEE International Conference on Information Acquisition*, pages 6 pp.+, 2005.

[14] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632, September 1999.

[15] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998.

[16] David M. Kristol. Http cookies: Standards, privacy, and politics. *ACM Trans. Internet Technol.*, 1:151–198, November 2001.

[17] William T. Harding, Anita J. Reed, and Robert L. Gray. Cookies and web bugs: What they are and how they work together. *IS Management*, 18(3):17–24, 2001.

[18] Carlos R. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of www client-based traces. Technical report, 1995.

[19] Pang-Ning Tan and Vipin Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Min. Knowl. Discov.*, 6:9–35, January 2002.

[20] Martijn Koster. A standard for robot exclusion. `http://www.robotstxt.org/orig.html`, 1994. [accessed on 19/12/2011].

[21] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1:5–32, 1999.

[22] Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou. The impact of site structure and user environment on session reconstruction in web usage analysis. In *Knowledge Discovery and Data Mining*, pages 159–179.

[23] Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A framework for the evaluation of session reconstruction heuristics in web usage analysis. *Informs Journal on Computing*, 15:171–190, 2003.

[24] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.

[26] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, Inc., 2001.

[27] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[28] Andrew Moore Dan Pelleg. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734, San Francisco, 2000. Morgan Kaufmann.

[29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[30] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *J. Intell. Inf. Syst.*, 17:107–145, December 2001.

[31] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[32] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *International Conference on Data Engineering*, pages 3–14, 1995.

[33] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Extending Database Technology*, pages 3–17, 1996.

[34] Mohammed Javeed Zaki. Spade: An efficient algorithm for mining frequent sequences. In *International Conference on Machine Learning*, volume 42, pages 31–60.

[35] Jiawei Han, Jian Pei, Behzad Mortazavi-asl, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *Knowledge Discovery and Data Mining*, pages 355–359, 2000.

[36] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, and Helen Pinto. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *International Conference on Data Engineering*, 2001.

[37] J. Heflin, J. Hendler, and S. Luke. SHOE A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland, 1999.

[38] Emmanuel Blanchard, Mounira Harzallah, Henri Briand, and Pascale Kuntz. A typology of ontology-based semantic measures. In *EMOI-INTEROP*, 2005.

[39] Michael Ricklefs and Eva Blomqvist. Ontology-based relevance assessment: An evaluation of different semantic similarity measures. In *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems*, OTM '08, pages 1235–1252, Berlin, Heidelberg, 2008. Springer-Verlag.

[40] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19:17–30, 1989.

[41] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[42] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In *International Conference on Computational Linguistics*, 1998.

[43] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Computing Research Repository*, 1997.

[44] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[45] Arindam Banerjee and Joydeep Ghosh. Clickstream clustering using weighted longest common subsequences. In *In Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, pages 33–40, 2001.

[46] Thomas Eiter and Heikki Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34:103–133, 1997.

[47] Jan Ramon and Maurice Bruynooghe. A polynomial time computable metric between point sets. *Acta Inf.*, 37(10):765–780, 2001.

[48] Birgit Hay, Geert Wets, and Koen Vanhoof. Clustering navigation patterns on a website using a sequence alignment method. In *In Proceedings of 17th International Joint Conference on Artificial Intelligence*, pages 1–6, 2001.

[49] Weinan Wang and Osmar R. Zaïane. Clustering web sessions by sequence alignment. In *In Proceedings of the 13th international workshop on database and expert systems applications (DEXA 2002). Aix-en-Provence*, pages 394–398. Springer-Verlag, 2002.

[50] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[51] Lin Lu, Margaret H. Dunham, and Yu Meng. Mining significant usage patterns from clickstream data. In *WEBKDD*, pages 1–17, 2005.

[52] George Karypis. CLUTO - a clustering toolkit. Technical Report #02-017, November 2003.

[53] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Mach. Learn.*, 55(3):311–331, 2004.

[54] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry - An Introduction*. Springer, 1985.

[55] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. Technical Report 00-034, University of Minnesota, 2000.

[56] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[57] Alexander Strehl and Joydeep Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)*, 3:583–617, December 2002.

APPENDIX A

COMPUTER SCIENCE DEPARTMENT TAXONOMY

The concept taxonomy used in this study is given below with associated keywords. Due to lack of space, some repeating parts are replaced with "..." The full taxonomy can be found at[7]. Each indentation level represents the depth of concepts in the taxonomy. The root of the tree is the concept called `Thing` which is not shown here. For example, the depth of concept `Person` is 1, depth of `Worker` is 2 and the depth of `tcan` is 6. Sets of words next to each concept represent keywords associated with that concept. For details, see Section 3.3.

```
Person
  Worker
    Faculty
      Professor (#Professor)
        AssistantProfessor (#Assistant*Professor)
          akyuz (#Oguz*Akyuz) (#Ahmet #Oguz #Akyuz)
          tcan (#Tolga*Can)
          ...
          erol (#Erol*Sahin)
          karagoz (#Pinar*Karagoz) (#Pinar*Senkul)
        AssociateProfessor (#Associate*Professor) (#Assoc.*Prof)
          alpaslan (#Ferda #Alpaslan)
          bozsahin (#Cem*Bozsahin)
          ...
          isler (#Veysi*Isler)
          oguztuzn (#Halit*Oguztuzun)
        FullProfessor (#Full #Professor)
```

[7] http://www.ceng.metu.edu.tr/~sefa/msthesis/ontology.dat

```
     volkan (#Volkan*Atalay)
     ...
     toroslu (#Ismail*Toroslu)
     yalabik (#Nese*Yalabik)
   Lecturer (#Lecturer)
   PostDoc
   Doctor (#phd) (#doctorate)
    birturk (#Aysenur*Birturk)
    ...
    faruk (#Faruk #Tokdemir)
  Assistant (#assistant)
   ResearchAssistant (#research #assistant)
   TeachingAssistant (#teaching #assistant)
    okan (#Okan #Akalin)
    rusen (#Rusen #Aktas)
    ...
    alan (#Ozgur*Alan)
    bugra (#Bugra*Ozkan)
 Student (#student)
  UndergraduateStudent (#undergraduate #student)
  GraduateStudent (#graduate #student)
Publication (#publication)
 Article (#article)
  JournalArticle (#journal #article)
  ConferencePaper (#conference #paper)
 Book (#book)
 Manual (#manual)
 Periodical
  Journal (#journal)
  Magazine (#magazine)
 Proceedings (#proceeding)
 Specification (#specification)
 TechnicalReport (#technical #report)
 Thesis (#thesis)
  DoctoralThesis (#doctoral #thesis)
  MastersThesis (#master #thesis)
 UnofficialPublication (#unofficial #publication)
```

```
Work
  Course (#course)
    MustCourse (#must #course)
      ceng100 (#ceng*100)
      ceng111 (#ceng*111)
      ceng140 (#ceng*140)
      ...
      ceng491 (#ceng*491)
      ceng492 (#ceng*492)
    TechnicalElectiveCourse (#technical #elective)
      ceng210 (#ceng*210)
      ceng220 (#ceng*220)
      ...
      ceng498 (#ceng*498)
    ServiceCourse (#service #course)
      ceng200 (#ceng*200)
      ...
      ceng494 (#ceng*494)
    GraduateCourse (#graduate #course)
      ceng500 (#ceng*500)
      ...
    MSCENGwoThesisCourse
      ceng508_2
      ...
      ceng714_2
    MSSEwoThesisCourse
      se448
      ...
      se706
  Research (#research)
    ResearchLaboratory (#research #lab) (#research #laboratory)
      BioinformaticsLab (#bioinformatics #lab) (#bioinformatics #laboratory)
      MultimediaDatabase (#multimedia #database #research) ...
      ImageProcessing (#image #processing #lab) (#pattern #recognition #lab)
      ISL (#intelligent #system #lab) (#intelligent #system #laboratory)
      Kovan (#robotics #lab) (#robot #lab) (#robotics #laboratory) ...
      LCSL (#computational #studies #language) (#computational #study #language)
      ParallelProcessing (#parallel*processing)
    ResearchGroup (#research*group)
      ComputerGraphics (#graphics) (#visualization)
      DataMining (#data*mining)
      EvolutionaryComputing (#evolutionary)
      GridComputing (#grid*computing) (#grid #compute)
    ResearchAssociatedCenter (#research*center)
      Modsim (#modeling #simulation)
      SRDC
Schedule (#schedule)
Resources (#resource)
  ComputingServices (#computing*service)
  Documents
    StudentDocuments (#student #doc) (#student #documents)
    StaffDocuments (#private #staff #doc) (#private #document) (#staff #document)
    NewsArchive (#news #archive) (#anouncement)
    Seminar (#seminar)
CSTopic
  DataStructures (#stack #queue) (#tree*structure) (#hash) (#data*structure) ...
  Algorithms (#sorting #algorithm) (#search #algorithm) (#graph #algorithm) ...
  DiscreteMath (#proposition) (#predicate #logic) (#set #theory) (#induction) ...
  TheoryComp (#theory #computation) (#automata) (#pushdown) ...
  PL (#programming*language) (#functional*language) (#object #oriented) ...
  OS (#operating*system) (#process #thread) (#deadlock) ...
  Digital (#circuit #digital) (#register #memory) (#arithmetic #logic #unit)...
  AI (#artificial*intelligence) (#heuristic #algorithm) ...
  Graphics (#computer #graphics) (#geometry #transformation) (#render #graphics) ...
  NLP (#natural*language) (#natural #language #processing) (#morphology) ...
  Database (#database #management #system) (#relational*algebra) (#sql) ...
  SoftwareEng (#software #engineering) (#project #management) ...
  PatternRecognition (#pattern #recognition) (#pattern #classification) (#bayes) ...
  ParellelComputation (#parallel #computing) (#parallel #computation) ...
  Bioinformatics (#bioinformatics) (#microarray) (#sequence*alignment) ...
```