USING ONTOLOGY BASED WEB USAGE MINING AND OBJECT CLUSTERING FOR RECOMMENDATION

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

BY

HAKAN YILMAZ

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

May 2010

Approval of the thesis:

USING ONTOLOGY BASED WEB USAGE MINING AND OBJECT CLUSTERING FOR CONTENT RECOMMENDATION

submitted by **HAKAN YILMAZ** 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 _____
Chair of Department, **Computer Engineering**

Asst. Prof. Dr. Pinar Senkul _____
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Asst. Prof. Dr. Osman Abul _____
Department of Computer Engineering,
TOBB University of Economics and Technology

Asst. Prof. Dr. Pinar Senkul _____
Supervisor, Computer Engineering Dept., METU

Asst. Prof. Dr. Tolga Can _____
Computer Engineering Dept., METU

Asst. Prof. Dr. Sinan Kalkan _____
Computer Engineering Dept., METU

Instructor Dr. Aysenur Birtürk _____
Computer Engineering Dept., METU

Date: 05.5.2010

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 : Hakan YILMAZ
Signature :

ABSTRACT

USING ONTOLOGY BASED WEB USAGE MINING AND OBJECT CLUSTERING FOR RECOMMENDATION

Yılmaz, Hakan
M.Sc. Department of Computer Engineering
Supervisor: Asst. Prof. Dr. Pınar Şenkul

May 2010, 116 pages

Many e-commerce web sites such as online book retailers or specialized information hubs such as online movie databases make use of recommendation systems where users are directed to items of interests based on past user interactions. Keyword-based approaches, collaborative and content filtering techniques have been tried and used over the years each having their own shortcomings. While keyword based approaches are naive and do not take content or context into account collaborative and content filtering techniques suffer from biased ratings, first item and first-rater problems. Recent approaches try to incorporate underlying semantic properties of data by employing ontology based usage mining. This thesis aims to design a recommendation system based on ontological data where web pages are seen as objects with attributes and relations. Instead of relying on users' content ratings, user sessions are clustered on a

semantic level to capture different behavioral groups. Since semantic information is used for the clustering distance function, each cluster represents a behavior group instead of simpler data groups. New users are then assigned to individual clusters that best represent their behavior and recommendations are generated accordingly. In this thesis we use the recommendation results as a means for measuring the effectiveness of the clusters we have generated. We have compared the results obtained using the ontological data and the results obtained without using it and shown that semantic integrating semantic knowledge increases both precision and recall.

Keywords: Web usage mining, Ontology, Clustering, Semantics Information

ÖZ

VARLIKBILIM TEMELLİ AĞ KULLANIM MADENCİLİĞİ KULLANARAK SAYFA TAVSİYESİ

Yılmaz, Hakan
Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Asst. Prof. Dr. Pınar Şenkul

Mayıs 2010, 116 sayfa

Günümüzde web üzerinden kitap satışı yapan birçok e-ticaret web sitesi ve internet film veritabanları gibi birçok bilgi merkezi web siteleri eski kullanıcıların davranışlarını baz alarak yeni ziyaretçilerini tavsiyelerle yönlendirmeye gayret etmektedir. Geçmişte kelime-tabanlı yaklaşımlar, işbirlikçi ve içerik tabanlı filtreleme yöntemleri kullanan tavsiye sistemleri kullanılmıştır. Bu sistemlerin herbirinin kendine has eksileri bulunmaktadır. Kelime tabanlı sistemler içerik ve durumu göz önüne almadığı için naifken, içerik ve işbirliği tabanlı filtreleme yöntemleri şişirilmiş değerlendirme, ilk içerik, ve ilk oylayan sorunlarıyla karşıyadır. Son zamanlarda ortaya çıkan yöntemler ise varlıkbilim yapılarını kullanarak sitemleri oluşturan etmenlerin anlamsal ve içeriksel özellikleri üzerinden giden ağ kullanım madenciliğine ağırlık vermektedir. Bu tez nesneler, nesne özellikleri ve nesneler arası bağları kullanan varlıkbilim temelli bir tavsiye

sistemi ortaya çıkarmayı amaçlamaktadır. Farklı kullanıcıların veriler üzerinde yaptığı değerlendirmelerden yola çıkmak yerine kullanıcıların geçmiş davranışlarını anlamsal seviyede gruplayarak ortaya davranış grupları çıkarılmaktadır. Basit veri gruplaması yerine varlıkbilimsel yöntemler kullanılması ortaya çıkarılan grupların davranışsal temellere sahip olmasını sağlamaktadır. Böylece yeni kullanıcılar en benzer davranış gruplarına atanarak tavsiyeler buna göre ortaya çıkarılmaktadır. Kullandığımız tavsiye sistemi bu tezde elde edilen kullanıcı gruplarının geçerliliğini test etmek için kullanılmaktadır. Bu çalışmada varlık bilim kullanılarak ve kullanılmadan elde edilen sonuçlar karşılaştırılmış ve varlık bilim kullanımının daha iyi sonuçlar elde edilmesine izin verdiği gözlemlenmiştir.

Anahtar kelimeler: Ağ kullanım madenciliği, Varlıkbilim, Gruplama, Anlamsal

DEDICATION

To My Family

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor Asst. Prof. Dr. Pınar Şenkul for her guidance, advice, criticism, encouragements and patience throughout the research.

Also we thank Süleyman Salın for providing some of the data used in this research.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Number of internet users has grown considerably over the past decade and continues to increase. Along with the number of users, data available on the internet continues to increase exponentially. Rapid growth of users of internet has given rise to e-business applications. Amazon/e-bay like online retailers have proven to be a convenient way to purchase items and have them delivered and so their revenues have shown great increase over the years. Of course given the success of those retailers many online shows have been driven to service of user where items from electronic appliances to books are now on sale on web sites. The number items presented in these sites makes it cumbersome for users to locate the items of interest within the site.

Not only e-commerce sites but public and free sites with thousands of pages like IMDB or library catalog web sites also face the difficulty of organizing and presenting the information to their users in a logical manner. It has become increasingly important for web site owners to direct their users to items within the site with as little effort from the user as possible.

So main reasons why web usage mining is motivated are

- To present users with relevant results and items as fast as possible (in the form of recommendations)
- By doing so, to have an edge in the competitive market
- By doing so, to decrease network traffic by letting users find what they are looking for by fewer page accesses

- Finding user behavioral patterns and exploiting them to increase sales/provide better service/providing better information

This thesis aims to detail a work that devises a recommendation system using web usage mining techniques while incorporating semantic knowledge into it. Web usage mining goes beyond keyword or single user approaches and answers more global questions about the behavior of groups of users.

Various methods and algorithms have been developed to provide reasonable recommendations to users ranging from simpler keyword based frequency counters to collaborative/content filtering techniques. Integrating context and content of the system and user data have shown increases in accuracy of the systems. In recent years there have also been some studies where semantic knowledge systems are mixed in to provide better results.

In all the methods used however the core of the system has always been the identification of the pattern a user shows while browsing a web site. Most service web sites have a well defined context in which they operate and provide service. Although all people are unique individuals, when many people access a certain resource some patterns emerge showing distinct group behaviors. For example an online bookstore caters to the needs of both doctors and lawyers people. When browsing habits of these groups are studied it is seen that lawyers tend to browse law related books more often while doctors browse medical resources more often than lawyers. If it was possible to simple identify such users by simply looking at their browsing data, it would be easier to recommend items of interests with a better accuracy, which is the main scope of this thesis.

In this work we try to identify user groups of an anonymous electronic book store by looking at the server logs. We evaluate the success of these identifications by mapping new users to the predefined groups and checking if the recommendations we provide accordingly are accurate. Although this approach

has been taken before by other studies we use some new methods to set this work apart and provide some contributions.

The first major point of the study is the usage of semantics. We have developed an ontology of the book store and while trying to identify user groups we do so by processing user sessions with regard to the underlying semantic elements of the books the user is accessing. Most other studies try to identify user sessions by simply looking at the page names the users access. There have also been studies where the page content (words) is taken into account. However by using the semantic power of the ontology we try to take content integration one step further.

The second major point is the integration of clustering techniques into this work. Clustering is a well studied and widely used technique to identify patterns in user groups. It has been used in other areas of works extensively. Clustering usage in web usage mining area however has been limited. Some studies have used pageview (page-id) clustering however using clustering in conjunction with semantics is a contribution of this work.

While identifying users from the web page access sessions most studies have ignored the order of the pages in which they were accessed. The last major point of this work is to identify user sessions as they are, a sequence of web page accesses, instead of simply a set of accesses. This ability will allow the system to acknowledge that certain page accesses may trigger others, providing better clustering.

The rest of the thesis will be organized as follows

- Chapter 2: Explanation of semantic web, its concepts, ontologies and OWL which is the current accepted language for ontologies.
- Chapter 3: Previous work in the area of web usage mining that utilize clustering and semantics.
- Chapter 4: Details of the work done by author and conclusion.

CHAPTER 2

BACKGROUND

This chapter will detail the background information, concepts and work related to web usage mining, semantic web and clustering.

2.1 Introduction To Web Usage Mining

This chapter gives an explanation of what web usage mining is, its roots, methods and an overview of the web usage mining process.

2.1.1 What Is Data Mining?

Web usage mining is a subset of web mining operations which itself is a subset of data mining in general. The aim is to use the data and information extracted in web systems in order to reach knowledge of the system itself.

To better understand the concepts brief definitions of keywords can be given as [1]:

Data: "A class of information objects, made up of units of binary code that are intended to be stored, processed, and transmitted by digital computers"

Information: "is a set of facts with processing capability added, such as context, relationships to other facts about the same or related objects, implying an increased usefulness. Information provides meaning to data"

Knowledge: "is the summation of information into independent concepts and rules that can explain relationships or predict outcomes"

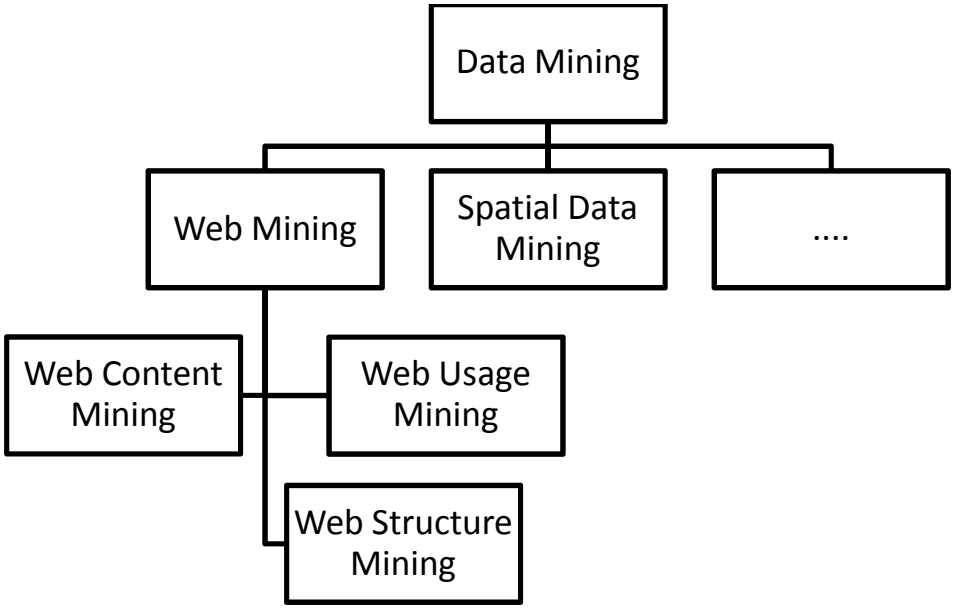

Figure 1- Figure Classification of Mining

2.1.1.1 Data Mining

Data mining is a set operations performed on a collection of data or a subset of it so as to extract meaningful patterns on the data. Another definition is "Data mining is the semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data" [2]. That is, data mining attempts to extract knowledge from data. If a subset is to be used, careful and unbiased sampling algorithms should be used to avoid biased result.

Data mining is different from information extraction although they are closely related. Information extraction is the process of extraction information from data sources whether they are structured, unstructured or semi-structured into

structured and computer understandable data formats. Data mining operations are performed on the data already extracted by means of information retrieval.

Based on the types of data sources it is applied on, data mining can be categorized. One such application would be on geographic data such as digital maps as usually seen most GIS applications which is called spatial data mining. Another area where data mining is widely used is bioinformatics where very large data about protein structures, networks and genetic material is analyzed. The sub category of interest in this thesis is the web mining which acts on the data made available in the World Wide Web (WWW) data servers.

2.1.2 Web Mining

Web mining consists of a set operations defined on data residing on WWW data servers. Mobasher et al. [3] defines web mining as "…the discovery and analysis of useful information from the World Wide Web". Such data can be the content presented to users of the web sites such as hyper text markup language (HTML) files, images, text, audio or video. Also the psychical structure of the web sites or the server logs that keep track of user accesses to the resources mentioned above can be targets of web mining techniques.

Web mining as a sub category of data mining is fairly recent compared to other areas since the introduction of internet and its widespread usage itself is also recent. However, the incentive to mine the data available on the internet is quite strong. Both the number of users around the world accessing online data and the volume of the data itself motivate the stakeholders of the web sites to consider analyzing the data and user behavior.

Web mining is mainly categorized into two subsets namely web content mining and web usage mining [3]. While the content mining approaches focus on the content of single web pages, web usage mining uses server logs that detail the past accesses to the web site data made available to public. Usually the physical structure of the web site itself which is a graph representation of all web pages in

the web site is used as a part of either method. However recent approaches [4] that appoint more focus on the physical link structure of the web site have introduced web structure mining as a separate concept. In order to understand the differences a brief description and area of work for each category is summarized below.

2.1.2.1 Web Content Mining

"Web content mining describes the automatic search of information resources available on-line." [5] The focus is on the content of web pages themselves. Mobasher [3] categorizes content mining as agent-based approaches; where intelligent web agents such as crawlers autonomously crawl the web and classify data [6] and database approaches; where information retrieval tasks are employed to store web data in databases where data mining process can take place [7].

Most web content mining studies have focused on textual and graphical data since the early years of internet mostly featured textual or graphical information. Recent studies started to focus on visual and aural data such as sound and video content too.

2.1.2.2 Web Structure Mining

One of the most well known algorithms, Page Rank Measure [8] and Hubs and Authorities [9] are based on the links between pages. Web structure mining focuses on the *links* rather than the content of the pages, their usage or semantics. [10] divides links into two categories. The hyperlinks that link the web pages and the document structure itself such as the xml or html structure. [5] details the latter.

2.1.2.3 Web Usage Mining

The main topic of this thesis is the web usage mining. Usage mining as the name implies focus on how the users of websites interact with web site, the web pages visited, the order of visit, timestamps of visits and durations of them.

The main source of data for the web usage mining is the server logs which log each visit to each web page with possibly IP, referrer, time, browser and accessed page link. Although many areas and applications can be cited where usage mining is useful, it can be said the main idea behind web usage mining is to let users of a web site to use it with ease efficiently, predict and recommend parts of the web site to user based on their and previous user's actions on the web site.

2.1.3 Overview Of Web Usage Mining And Recommendation Process

All approaches to web usage mining have some basic steps. They are

- Preprocessing
    - Acquisition of data
    - Cleaning data
    - Identifying page views
    - Identifying Users
    - Identifying sessions
    - Path Completition
- Mining the Data
- Recommending URLs/Presenting Analysis results

The first phase is mostly identical in all systems which is the preprocessing phase. Of course the mining and recommendation phases are seen to be different in different systems.

An overview of jobs done in the preprocessing phase are given below.

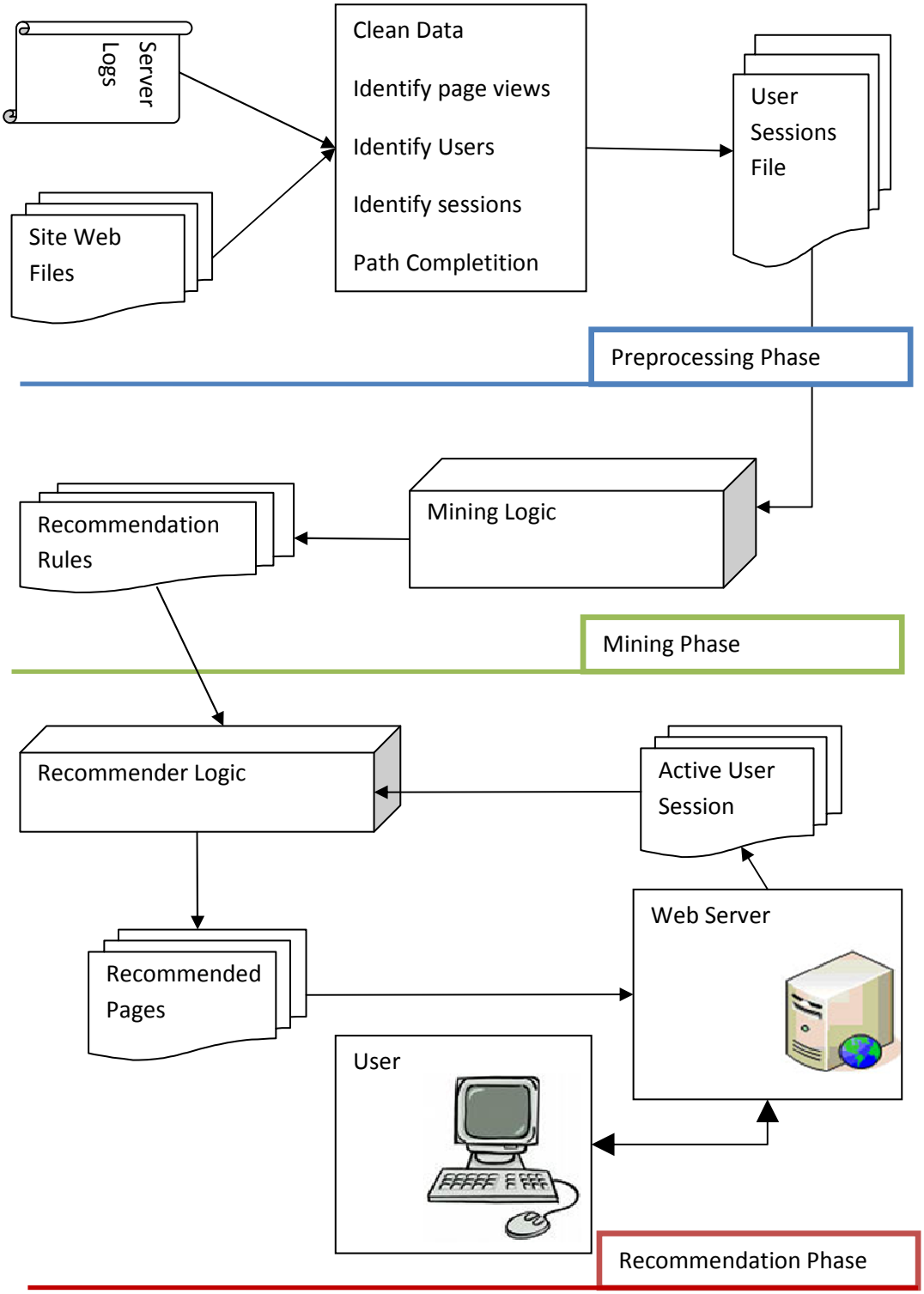

Figure 2 - An overview of a Recommender System

2.1.3.1 Preprocessing

2.1.3.1.1 Acquisition Of Data

Web usage data can be acquired from three sources; server logs, clients and proxies [11].

One source of web usage data are the server logs such as IIS or Apache. Since all web servers have some sort of logging capability any web host is capable of providing a log file. The size of the log file will be determined by the time frame logging was in action, number of user actions on the web site and the average number of page accesses by user in each session. If too little a time is selected to get a log dump from server the resulting data can be too little to be of any real use. If a too large time frame is used the data could be too much to manage efficiently without providing any extra benefit over a smaller log. So according to number of page accesses per hour, a sufficient time frame should be chosen to take a server log dump.

A point of consideration could be the specific temporal issues related to usage of a web site. For example if an e-commerce site is known to sell certain products frequently on certain days of week it could be beneficial to take samples of data each day of week and aggregate them into a larger dataset so as not to bias the recommendation engine.

Another possible data source is special software running on client machines. This requires a web site to install the software on each user accessing the web site and let the software collect user navigation data. After some time the software in each user's machine reports back the collected data and user sessions are generated. Since this method requires every user's consent to installing the software it is very difficult realize in generic web sites thus rendering the method largely ineffective.

One last way of acquiring the data would be to get the server logs on proxy servers. This could be used to track usage data across many web sites which is beyond the scope of this thesis.

So we will focus on simply HTTP server logs through this thesis. The format of the data made available in the server log can be Common Log Format (CLF) or Combined Log Format. Combined format is the same as CLF except it has two more fields at the end. A sample is analyzed below (apache documentation [12]):

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"

127.0.0.1

This is the IP address of the client

-

The "hyphen" in the output indicates that the requested piece of information is not available. This field is not used.

frank

this is the userid of the person requesting the document as determined by HTTP authentication. Unused if the page is not password protected so mostly unused by web usage parser.

[10/Oct/2000:13:55:36 -0700]

The time that the server finished processing the request.

"GET /apache_pb.gif HTTP/1.0"

The request line from the client is given in double quotes. This is the web resource the user is trying to access. In this case a .gif file.

200

This is the status code that the server sends back to the client. This code can be used to find out incorrect request in the data cleaning step.

2326

The last entry indicates the size of the object returned to the client, not including the response headers. It is mostly irrelevant for our purposes.

http://www.example.com/start.html

This gives the site that the client reports having been referred from. It is mostly irrelevant for our purposes.

"Mozilla/4.08 [en] (Win98; I ;Nav)"

The User-Agent HTTP request header. Again it is mostly irrelevant for our purposes.

2.1.3.1.2 Cleaning The Data

There are multiple image files in pages that are not part of the content at all. For example images on buttons on the web page or graphical parts that cover some parts of the page to give it a certain feel and look are of this type. These images need to be filtered out.

In this phase multiple frame web pages also need to be handled. Also dynamic content present in the web page needs to be taken into consideration.

Handling invalid requests is a part of cleaning phase. For example if a user attempts to access a web page that is not part of the web site the web server will return a code to warn the browser which is called the status code. The status code for “OK” which is returned for successful operations is 200. Other codes can be

found in the HTTP specifications [13]. Filtering non 200 requests is a fast, easy and efficient way of filtering problematic requests.

Many web crawlers make frequent visits to web sites to log the pages for search algorithms. Since we are not interested in crawler activity but rather human accesses these accesses need to be pruned from the web log. [14] Describes a method to do that.

As the last step of data cleaning phase non-existing pages need to be handled. The page could have been served at some point to a user but that does not necessarily mean it is still present among the available pages. So the current web site topology should be considered and cross-checked to see if the served page is still present.

2.1.3.1.3 Identifying Page Views

When a user makes page request from a server most of the time that request arrives at the server as a group of resource requests. The reason is that each web page usually consists of different parts such as image files, audio, video, textual information or even multiple frames of pages. Each of these resources are explicitly expressed in the server log file.

All of this data may be useful for different applications. For example a multimedia mining application could be specifically interested in the graphical or aural data pointed in the server log. Also any form of classifier that can take this type of data into account while classifying these pages would be interested in them. However for our purposes we are interested in mostly the textual information present in files. So we will filter out any lines of requests with extensions such as .gif .jpg .mp3 .avi etc.

2.1.3.1.4 Identifying Users

Most web sites do not force users to register to view content. So when a user access is logged we most often do not have any way of uniquely identifying different users. What we have in the log file that can be used as identifiers are

- *IP address:* This is the IP address where the request has come from. Unfortunately IP addresses are not unique keys assigned to a single person on the planet.
- *User agents:* This identifies the user browser that requests the access. Again multiple users can use the same browser agent so this is not a unique identifier.
- *Referrer:* Although knowing where the request is directed from does give some heuristic maneuverability it is not enough to identify users.

[11] summarizes these issues as:

a- Single IP address/Multiple server sessions: This is the preferred case.

b- Multiple IP address/Single server session: Randomly changing IP's at every request

c- Multiple IP address / Single user: IP of a user changes at each connection reset. This is seen with dynamic IP address users.

d- Multiple agent/Single user: Browsing the web with many browser agents.

Also note that some applications turn the IP addresses into geographic locations by reverse DNS lookup to generate user/location pairs.

Once these issues are handled we have a cleaned out server log file where each access is labeled with a user identifier.

2.1.3.1.5 Identifying Sessions

After identifying the users we have a list of page accesses with user ID's attached. In this step we sort this list first according to user ID's and then according to access times. However each group of accesses with the same user ID do not necessary belong to the same session. For example a user might visit the web site at 13:00 and access some pages, and then at 15:00 visit the site again and access some more pages.

So these accesses should be divided into separate sessions. Experiments show 20-30 minutes act as a good session breaker limit.

2.1.3.1.6 Path Completition

When a user presses the "back" button on the web browser, most of the time the browser do not generate a page access request to server but reloads a previously cached version of the page. This is done to improve page access speeds when possible. However that means some of the page accesses by the user are not recorded in the web access log files.

Another caching mechanism is present in the proxy servers. When a page access is requested the proxy may decide the local proxy cached version is no different than the page on the server and serve the cached page instead. Again this access is not logged in the server file.

Consider the simple example topology shown on Figure 3. If a user access A->B and presses the "back" button and goes to *c* the server will log this session as A->B->C.

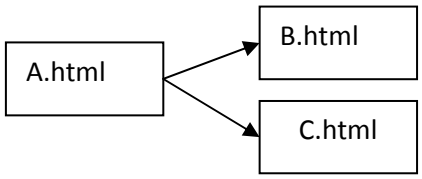

Figure 3 – Sample Taxonomy

The intermediate *A* is omitted and a non-existent link between *B* and *C* is assumed. This is why path completion is needed. Analyzing the access log and the topology of the site imaginary accesses may need to be created to gain a consistent access log.

At the end of this stage we now have structured document that can be grouped by users, sessions and access times.

2.1.3.2 Mining The Data

At this stage data mining algorithms are employed on the preprocessed data to discover interesting rules, associations and facts about the way users interact with web site. Although we are specifically interested in web usage mining it should be noted that any data mining technique could be viable in this step since web usage mining itself is a data mining operation applied on web server logs. [15] gives four techniques as the most commonly used ones in the area of web usage mining.

2.1.3.3 Statistical Analysis

Ever since the advent of internet there have been tools to analyze the server logs and new and better ones keep emerging [16-19]. There is a strong statistical

background behind such work however the tools are rarely a result of academic research but rather results of simple business needs.

These tools give detailed information about the way a web site is used however lack the ability to produce behavior patterns or an in depth analysis of why such a usage pattern exists. Statistical methods include counting, histograms and probability. The results of log analysis by such tools can be presented in graphs, charts or tables (Figure 4).

Possible benefits could be

- Show unused/inaccessible web site parts. It is possible that some specific part of the web site is not or rarely if ever accessed.
- Show frequently used web site parts. Parts of the web site could be rearranged to even out the traffic or such pages could be improved for improved customer perception.
- Based on user identification part of preprocessing geographical web access information could be identified.
- Simple information like average access times, average page access numbers, daily hit counts can also be extracted at by statistical approaches.

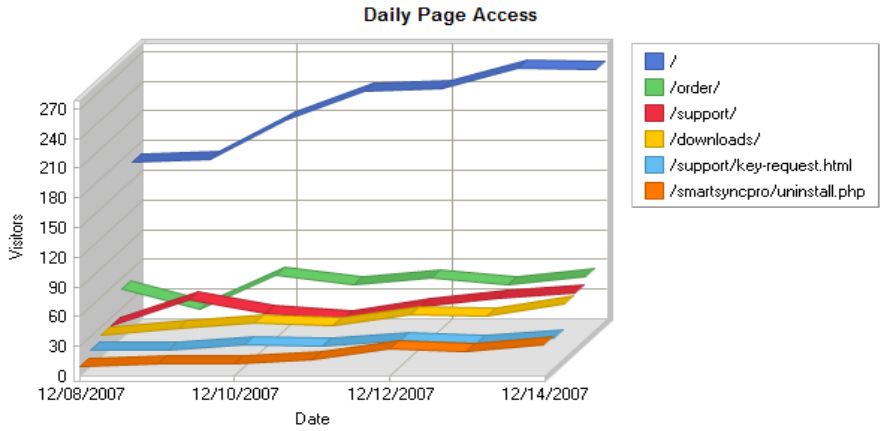

Figure 4 - A graph view of daily page accesses

2.1.3.4 Association Rules

This is the most popular form of unsupervised data mining technique used today by researchers and application developers. The technique is intuitive and gives a good balance between usefulness and implementation complexity

Association rules have the form of

IF eventX then eventY %T of the time in %Z of cases

Example: fish =>chips %35 %10

The first part of the rule is called the "antecedent" and the second part is called the "consequent".

Events: Here the eventX and eventY can be viewed as page accesses.

Confidence (Accuracy): T is a percentage of times this rule is applied i.e. when eventX happened eventY also happened.

Support (Coverage): Z is the number of times this event was observed in the whole space of events where the consequent took place.

Both the antecedent and the consequent can consist of more than one item thus making association rule mining a viable candidate for recommender systems. We can have rule stating {A, B, C }➔ {D, E} and if a user actually visits A, B and C it will be possible to recommend D and E based on the support and confidence values associated with the rule.

The rules generated with association mining are usually too many to be of any use so usually such systems need to define how to extract "Interesting Rules" from such systems. That of course is mostly dependent on how the rules will be used. [20] defines how these rules can be used to extract useful information.

a- **Target the antecedent:** This can be used to analyze the impact of an item on the rest of the system. For example to analyze if removing page A.html from the web site is a good idea or not we can extract all rules with A.html as antecedent and see if effects any important part of the site.

b- **Target the consequent:** This approach can be used to find out what effects the usage of the consequent. For example we can find out which pages lead customers to a certain web page B.html in our site by analyzing the rules having B.html as consequent.

c- **Target based on accuracy:** Sometimes we more interested in the fact that some certain event occurs more than how often it occurs. For example an e-commerce site might be more interested in a less frequent case where purchase of an item X almost always triggers a purchase of item Y which is quite profitable and want to exploit such situations.

d- **Target based on coverage:** Database administrators are usually interested in optimizing most frequently used parts of their systems. So getting rules with very high coverage can be utilized for such intentions.

Based on the usage scenarios the accuracy and support thresholds are usually given to prevent generating too many rules. It is nearly impossible to find any useful rules with both high support and accuracy so a tradeoff has to be made. Applications usually start with a minimum support count of 50% to prune very simple rules from start.

Another problem is seen while generating aggregated rules. For example the rules

A➔B 0.7 0.2

C➔B 0.5 0.8

Can be combined into

{A, B} ➔ C

and a resulting support and accuracy needs to be found. However it is not clear from inspection whether the values should be summed, multiplied or some other mathematical operation performed. It depends on the methods of the application to determine the accuracy and support of such rules.

Association rule mining is a well known and researched area of data mining. [21] presents the idea of strong rules in 1991. In 1993, [22] presented a method for discovering association rules between items in market databases using basket analysis. From the formal definitions of [22]:

Let $I = i_1, i_2, i_3, \ldots i_m$ be a set of items in a data space.

Define a transaction as $t = t_1, t_2, t_3, \ldots t_k$ where t[a] is 1 if transaction contains item i_a and 0 otherwise.

Assume we have a database T of such transactions.

An association rule is defined as $X \rightarrow I_j$ where X is a set of items in I.

Main interest is to find rules that have a minimum support count (minsup) so that generated rules have statistical meaning. Also these rules are required to offer high confidence percentages so that they are usable as business information. Agrawal [22] suggested that the support of a rule X➔Y is the same as support of X. Confidence of this rule is the ratio of the supports of X➔Y and X. Rest of the proposition deals with memory management and processing speed issues based on these definitions.

Table 1 - A representation of items as baskets

Customer_id	Item_ID
1	1,5,10
2	2,8
3	1,3,9,12

The article attracted a lot of attention. In the following years many articles were published that refined or progressed the idea [23-27]. Since the problem domain usually has item sets of millions, most effort was spent on improvements that either decreased database load such as the Partition Algorithm [26] decreased memory usage by employing apriori variants or decreased rule generation run-time by employing sampling techniques on data [27].

2.1.3.5 Sequential Patterns

Sequential pattern generation is an extension of association rule generations. While in association rule generation we deal with rules as form

(A set of Events) ➔ *(Another set of Events)*

Sequential patterns add a temporal factor into the rules

(A time ordered list of Events) ➔ *(Another time ordered list of Events)*

Table 2 - A representation of items with timestamps

Customer_id	Timestamp	Item_ID
1	02/08/2005	3
1	02/10/2005	9
2	02/08/2005	1,2
2	02/18/2005	3
2	02/28/2005	4,6,7
3	02/08/2005	3,5,7
4	02/07/2005	3
4	02/12/2005	4,7
4	02/15/2005	9
5	02/19/2005	9

For example while applying association rule generation technique to supermarket cashier log data we have sets of item purchases. A customer could pick any item from the shelves in the market but while the items go through the register that order is lost. However if the same system was applied to an online marketing system we could have logged each item with a timestamp as they make it into the virtual shopping basket.

In the case of server log parsing we are presented with page access time stamps that help us order page views according to the time they were requested. This extra information enables the recommender systems to take the order of previous user page views into consideration and present better recommendation result to users.

The sequential nature of page accesses allows the system to generate rules such as

A->B ➔C and B->A➔D

with higher accuracy than the non-temporal counterpart where A->B and B->A were both expressed as {A,B}

Agrawal et al. [28] extended their study on association rule generation by including timestamps into the equation. The problem statement is "given a database of transactions where a transaction consists of a user_id, *timestamp* and a set of item". Note that a transaction is still defined as a "set of items". So there is not any ordering in a single transaction.

The idea is to capture the behavior of customers that span over a period of time. For example a customer can buy some books from a book store. When next time the same customer comes to the store we expect him to choose to buy new books based on the previously bought ones. For example if he bought Lord of the Rings: Fellowship of the Ring the first time, we may expect him to buy Lord of the Rings: The Two Towers the second time. And we expect that such temporal behavior is present in the set of whole transactions in that book store.

In formal definitions an *itemset* is a non-empty set of items, denoted by i = (i_1, i_2, i_3… i_m) where each i_j is an item. . A *sequence* is an ordered list of itemsets denoted by < s_1, s_2, s_3… s_n> where s_k is an itemset. A sequence s1 = < a_1, a_2, a_3… a_m> is a subset of another sequence s2 = < b_1, b_2, b_3… b_n> if there exists a set of items < i_1, i_2, i_3… i_k> in s1 where each i_j has a corresponding order in s2. For example <(3)-(4,6)-(8)> is a subset of <(1),(2,3),(4,5,6),(8)>. Any sequence which is not a subset of another is said to be *maximal*.

Furthermore a *customer sequence* is defined as the ordered set of a single customer sessions merged together. For example the customer sequence of customer2 in Table 2 is <(1,2) (3) (4,6,7)> . A customer supports a sequence s if s is a subset of the customer sequence. The *support* of a sequence is the fraction of customers that support it. [28] gives three algorithms to tackle the problem namely AprioriAll, AprioriSome and DynamicSome.

Later [29] extended the problem definition by adding taxonomies and time windows into the equation but the original problem statement remained the same. In that article a GSP (Generalized Sequential Pattern) algorithm is devised which is handles the new constraints and is up to 20 times faster than the AprioriAll. In [30] a new algorithm called PSP, which is largely inspired by GSP, was proposed that showed some memory usage improvements over GSP and shorter execution times as minsup constraint was tightened. Although the ideas in GSP and PSP make use of taxonomies it should be noted that a taxonomy is strictly a is-a hierarchy and is not an ontology. SPADE [31] was introduced later on improving on older algorithms. It uses a vertical ID list structure, makes use of equivalence classes and makes only three database scans. First scan finds frequent items of length 1, second scan find frequent items of length 2 and last scan finds the rest. However as common in older algorithms it suffers from the high number of candidates generated in first and second step especially and is inefficient in mining longer patterns.

[32] proposed a new algorithm called FreeSpan which has speed improves over older algorithms and also has a more linear execution increase plot as number of sequences to parse increases. [33] proposes improvements over FreeSpan and proposes a new algorithm called PrefixScan. Both FreeSpan and PrefixSpan algorithms are pattern growth algorithms as opposed to frequent item counting. PrefixSpan is much faster and more efficient in capturing time-constraints. The efficiency of PrefixScan comes from the fact it does not need to generate candidates as seen in GSP type algorithms. One of the latest algorithms in the span family is CloSpan [34] which is based on graph theory and produces only closed sequential patterns. "A closed sequential pattern *s*: there exists no superpattern *s'* such that *s'* ⊃ *s*, and *s'* and *s* have the same support". This reduces the number of generated patterns however it is possible to generate all possible patterns from the produced set. CloSpan has speed improvements over other algorithms and can generate longer sequences of patterns.

Integrating temporal data into usage mining approaches makes it possible to make predictions into near future about user behaviors, and observing common past behavior based on recent actions, answering questions such as

- What would a user who bought bookX buy in the next month?
- What have customers who buy itemX today bought in the last week?

In the context of web usage mining items are considered as pages and itemsets are sessions. From the problem definition it can be seen more than one session per user is recorded. In some applications where users are uniquely identified such as online retail sales this corresponds to each user transaction over a period of time. However in the more general case where it is not possible to identify users uniquely each session has to be assigned a new ID thus turning this approach into simpler association rule mining. However it is possible to apply the temporal logic to pageviews instead of recording them as simpler sets without

timestamps. By doing this conversion it is possible to convert the web log mining problem into sequential pattern mining.

2.1.3.6 Clustering

Clustering is the technique of grouping together similar data. The idea is well studied and many techniques exist today. Since the method used in this study for generating patterns is also clustering, it is explained in a separate section.

2.1.3.7 Recommending Urls/Presenting Analysis Results

This step is where the result of data mining is utilized. For example if statistical analysis is used charts, tables and graphs are generated from mined data and presented to users of the system. If the system has a recommendation engine, the recommendation engine is supplied with active user session and data mining result for future prediction.

Any system that is to utilize mining results for recommendation needs to have a recommendation engine to handle the specific data output of mining process and a way of presenting the recommendations. This phase is highly coupled with the mining phase so the recommender and miner engines usually work on similar principles and methods.

2.2 WWW For The Machines - The Semantic Web

Semantic web is a growing interest area and is envisioned to be the future of WWW. Our research aims to integrate semantic knowledge into web usage mining. In this chapter an overview of semantic web and related concepts are summarized.

2.2.1 What Is Semantic Web?

2.2.1.1 The Rationale For The Need

In 1989 Tim Berners-Lee [48] proposed a system which he called “Mesh” to CERN management where he claimed a global hypertext system where researchers could publish and manage data was in best interest of CERN. Later the CERN physicists were using the system and many academicians after that. In the past 30 years that system turned in World Wide Web that came to be used by billions of home computer users as the single largest data source available on world.

WWW, also known as W3, as we know and use today is a wide array of hyper tagged information residing in web page servers. Clients, known as browsers, process this information and display on computer screens according to the tags embedded in the documents. The protocol to make the server/client interaction possible is called the HTTP and is defined as “…an application level protocol for distributed, collaborative, hypermedia information systems” [49]. The specification agreed upon for marking the information so as to make it possible to present it in a more readable form is called the HTML which stands for Hyper Text Markup Language.

A simple HTML page is given below.

```
<html>
      <body>
      <h1>My First Heading</h1>
      <p>My first paragraph</p>
      </body>
</html>
```

Data embedded in the format is tagged between <> and </> where the tags tell any software to read this document where the heading of the document is where

paragraphs begin, what font and size to use etc. HTML specification has been revised and other languages have been introduced such as XHTML to introduce more functionality over the years however the descriptive nature of the languages mostly stayed within the limits of cosmetics. Although the data is embedded in the pages, the language does not tell us anything about the semantics of it. For example a web page can contain the name of the author of the page and specify that it should be rendered in bold and in font 12 but does not tell us that part of text is actually a human name.

As seen in the example this format is specifically prepared so that client browsers can parse the data and display it appropriately. The text is for human understanding. In order to make any use of this information a human needs to read the text. That was in fact the idea behind the WWW and we can say it has fulfilled the promise in that sense.

However since the data on the web has long gone beyond the size where humans can find, access and process it there is a growing need for machine-understandable data representation. When we say machine-understandable we do not mean a comprehension on human scale but rather the ability to categorize data on the web as objects, their relations, and properties and infer simple logical rules accordingly. This idea is called the semantic web.

Sir Tim Berners-Lee et.al who is now the founder and head of World Wide Web Consortium (W3C), an organization whose purpose is the development of standards for the World Wide Web, published another article in 2002 [50] where he envisioned a future where semantic web has widespread usage and thus "intelligent" agent can answer complex question by surfing the web, finding related information, linking them together and applying logic operations.

2.2.1.2 About Meaning

Semantics as a word means the "study of *meanings*" [51]. In the world of data representation we are interested in the *meanings* of words. When we think of

words we make associations of the word with other known words. For example when we know someone is a *teacher* we automatically associate words with it. We know she is *human, works* in a *school*, *teaches student*. We also consider that the *teacher graduated* from some *university, lives* at some *address.*

Of course since human mind is a large database of such concepts and their relations and attributes we easily associate many words with others thus comprehending the meaning of the world. However when a person with little prior knowledge of a subject is presented with a word from that subject that word has little meaning and is of little use. When physicist talks about relativity and time-dilation few, if any, people can associate a meaning to the words where a fellow physicist would *know* exactly what he is talking about.

What is meant by computers to understand data is exactly the same. "The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [50]. The idea is not make computers suddenly understand and talk like humans but rather tagging the prepared data in the web such that machines can now know *xpo23* is a human whose *name* is "Ali" *worksAt* a *school* which is *locatedIn "*Ankara*"*. Knowing this much would enable to extract "Ali" when a query is sent asking for all *humans* that *work* in "Ankara".

2.2.1.3 Data Representation

In order to make use of the data even if they are tagged to give meaning to it, the software has to know the format that data is prepared. Different data mining applications have used such data representation techniques in the past however in order to make use of the data prepared by other parties a common format needs to be defined. So a well defined format must exist in order to share the information over the web.

Moreover a mechanism needs to exist that will enable different systems with different descriptors to same concept to understand each other. “Traditional knowledge-representation systems typically have been centralized, requiring everyone to share exactly the same definition of common concepts such as "parent" or "vehicle."” [50]. However it is not possible for such a system to exist when the data medium in the WWW and the amount of data is well beyond the capabilities of organizing in a single very large structure.

One of the main benefits of traditional WWW has been that anyone from students to workers, scientists to housewives could and have been able to publish data on the web and link to any other resource over the web as well. The appeal of this freedom enabled W3 to grow exponentially over the last decades turning the W3 into a huge database. Of course this also means a lot of information in the web is unaccountable, false or simply intentionally misleading.

The idea behind the semantic web is to let people continue to publish data as easily as it was before only in a form that is machine readable. So restricting data representation to a centralized system would not be desirable even if it was possible. The idea of semantic web is to let systems compare their data in a way to identify common concepts and find out equalities so as to integrate one another. So the basic building blocks of semantic web will allow users to define their own types, attributes and relations without the need of a centralized authority.

2.2.1.4 Inference And Logic

One of the most powerful features of such knowledge representation is that, as more and more data is bound together with inheritance, attributes and relations (is-a, has-a, does) more knowledge can be extracted from already existing data by induction and deduction.

An example deductive reasoning is

No machine is 100% efficient (premise)

Toaster is a machine (premise)

Toaster is not 100% efficient (conclusion)

In this case without explicitly stating that a toaster is not 100% efficient we could logically arrive at the conclusion.

An example inductive reasoning is

Most machine work on electricity

Toaster is a machine

Toaster is works on electricity

In this case by observation we do know that most machines work on electricity and arrive at conclusion that a toaster would work on electricity too. That might be correct for the toaster but it could well be the case that the toaster worked on some other energy. This is an inherit problem in inductive reasoning. However we humans use it in everyday life quite often and usually arrive at correct conclusions. So it should be possible to code such rules into inference engines too.

The idea of inference is closely related to scientific method, which is a base for all science branches. So the subject is largely discussed in all fields. See [52][53] for a more complete analysis of the subject.

It should also be noted that since any user can and will publish their own types and relations in semantic web, there is no way to tell if the premises are indeed correct. For example a web site could say that a *dog isTypeOf Plant*. If inference rules are applied naively it can lead to wrong conclusions. [50]'s solution to problem is digital signatures. "Another vital feature will be digital signatures, which are encrypted blocks of data that computers and agents can use to verify that the attached information has been provided by a specific trusted

source". So the software to apply inference logic on untrusted sources would be cautious of the results obtained. However of course that does not mean human error while preparing data can always be prevented or that any source without a valid signature is wrong. So inference engines will still have to maintain a degree of logic to prune information that does not make sense.

2.2.2 Language Of Semantic Web

In order to incorporate semantic knowledge into web pages a new set of document formats and some new ways to represent data had to be invented along with using already existing formats and structures. For example XML syntax has already been used for data identification purposes for years. We are already familiar with URI's, one common example of which is the URL, as a resource locator over the web. [54] and [55] give a good introduction to semantic web topics. An abstract view of all technologies and languages that make up semantic web is given by Tim Berners-Lee [56] in Figure 5. In this section the concepts and languages making up the building blocks semantic web are explained.

2.2.3 Unicode

Semantic web is designed in a way to have the ability interconnect every data node on the web. So there should not be a language representation restriction on the system which is easily overcome by choosing Unicode for the base character set.

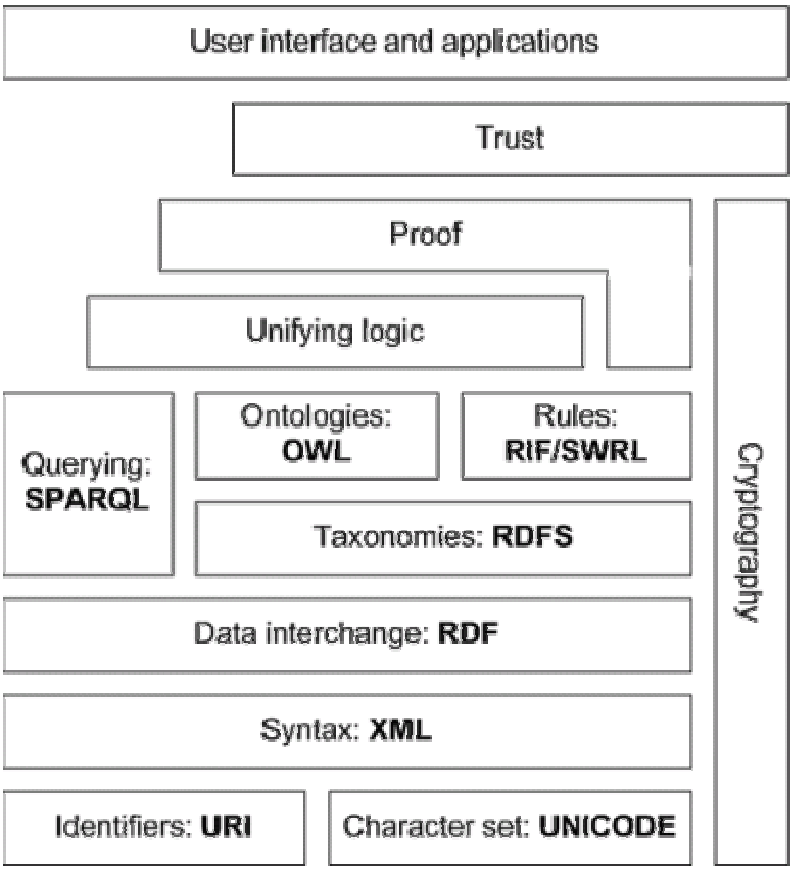

Figure 5 - Semantic Web Stack

2.2.4 Uniform Resource Identifier

In URI specification it is defined as "A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource" [57]. An URI is the way to identify anything that is on the web. It can be said to be basic building block of the web. If we want to reference anything on the web it has to have an URI, and anything can be given an URI. There are many URI schemas in use today. A comprehensive list can be found in the w3c site [58].

One of the most well known URI type is the URL. For example **http://www.metu.edu.tr** identifies the METU web site. URL's are also by means

of centralized DNS servers resource locators unlike most other URI's. URN for example only identifies the name of a resource. It should be noted that the job of an URI is not to locate the resource but to identify it. URL's manage this only because of DNS servers.

There is no centralization for URI's. That means anyone can define and create URI's. That also means that there is no central agency to control the ownership of URI's, their contradiction or owners. Although this gives great freedom for web designer it also means they need to be aware that URI's are not unique. Two identical URI can point to different resources or two different URI's can point to same item.

An example URI is tel:+1-816-555-1212 which simply identifies the numbers +1-816-555-1212 as a "tel".

2.2.5 Extensible Markup Language (Xml)

By definition from w3c who defines the XML specifications [59] "Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879)."

XML is a language that we can arbitrarily tag (markup) any arbitrary text. Any xml document is made of markups and content. Markups are either of the form <somemarkup> or &somevalue;. Anything that is not markup is content. For example the simple sentence

Roses are red

Can be expressed in XML as

```
<sentence>
        <plant>Roses</ plant > are <color>red</color>
</sentence>
```

Notice that the content is still the same. However now the computer can know that *Roses* is a plant and *red* is a color. Adding some attributes to tags

<sentence>
< plant type="flower" >Roses</flower> are <color code="xFF0000">red</color>
</sentence>

Now the computer knows *Roses* are not only plant but of type *flower*. Of course as with every system without a central identifier bank the identifiers in this XML document can get confused with another document. So XML introduces *namespace* concept just like found in most computer languages today. By defining a namespace using an UIR at the beginning of the document, we can uniquely identify our identifiers. Moreover XML provides a way to abbreviate the namespaces. Example plant namespace can be seen below.

<sentence
  xmlns="http://example.org/xml/documents/"
  xmlns:plant="http://plants.net/xmlns/"
>< plant :plant plant :type="flower" >Roses</ plant :flower> are < plant :color
plant :code="xFF0000"> red</ plant :color> </sentence>

2.2.6 Resource Description Framework

Abbreviated as RDF, resource description framework is a syntax framework designed to exchange information in a machine interpretable way. W3C defines RDF as "… a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web" [60].

For now we have a way to identify or locate resources in the form of URI's. We also have a language available (XML) which allows us to tag textual data. RDF's combine URI's using XML to describe objects, attributes and relations between objects. Of course all of this is done in a way that machines can process and "understand" this data.

The syntax of RDF's is of triplets where each member is an URI (or black). Subject->predicate->object and in that order. [61]

- the subject, which is an RDF URI reference or a blank node
- the predicate, which is an RDF URI reference
- the object, which is an RDF URI reference, a literal or a blank node

An example

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

This RDF defines

- default names space :rdf
- user defined namespace :contact
- A Person with
  - FullName = Eric Miller
  - Mailbox = mailto:em@w3.org
  - Title = Dr.

RDF allows interleaved descriptions. So a subject/predicate/object can also be a triplet. Also the syntax allows more than one attributes to be defined in a single RDF statement. So a single RDF statement can handle all the information we want to convey about a certain object which is powerful property.

2.2.7 Rdf Schema

The word schema originates from Greek; meaning shape, form or a plan as more general view. Schemata in computer world are usually description files about other files. This allows a certain abstraction of levels in definition of data as shape (how data is used) and content.

While RDF allows object representations, their attributes and properties it does not provide a mechanism to define hierarchy relations between objects or relations. This is the reason RDF schema is defined. [62] "The RDF data model, as specified in [RDFMS], defines a simple model for describing interrelationships among resources in terms of named properties and values."

RDFS is defined in "http://www.w3.org/2000/01/rdf-schema" so all terms of RDF schema start with this URI which is abbreviated by "rdfs:". RDFS has built in classes such as rdfs:Class, rdfs:Resource, rdfs:Datatype, rdfs:Property and rdfs:Literal. Since they are classes they are defined using Notation3 [63] as

```
rdfs:Resource rdf:type rdfs:Class .
rdfs:Class rdf:type rdfs:Class .
rdf:Property rdf:type rdfs:Class .
```

Which means Resource, Class and Property are all of type Class. Using the same notation it is possible create new classes.

```
:Car rdf:type rdfs:Class .
```

:Honda rdf:type rdfs:Car .

Two lines say that Car is Class and Honda is a Car which makes it implicitly a Class too. Note that Honda here is just the name of class. We could name the conceptual Honda cars as 3uoiu354* without consequence. These definitions are for machine interpretation so the computer would not know the actual Name property of class Honda is "Honda" too. Instead we can define

:name rdf:type rdf:Property .
:Honda :name "Honda" .

Now the computer also knows the name of Class Honda is also "Honda".

RDFS also defines some built-in Properties such as rdfs:range, rdfs:domain, rdfs:type, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:label and rdfs:comment.

The range and domain properties allow us to specify the domain and range of properties such as

:bookPrice rdfs:domain :Book .
:bookPrice rdfs:range rdfs:Literal .

The type, subClassOf and subPropertyOf properties allow definition of inheritance relations such as

:Rose rdfs:subClassOf :Flower .
:Flower rdfs:subClassOf :Plant .
:iff rdfs:subPropertyOf:imply

The label and comment properties are used for putting up comments on properties and classes such as

:Honda :name “Honda”
  rdfs:comment "Name of Honda"

A complete list of all vocabulary can be found in [64].

2.2.8 Ontologies

2.2.8.1 Concept And Definitions

Ontology is a Greek word meaning study/science (logy) of being (onto). It can be expressed as the study of being, existence and reality. It is a sub-branch of philosophy. The idea of ontology dates back to Parmenides, Aristotle and Plato.

Since ontology deals with existence of entities, their hierarchical properties and relations it has been a topic of interest for all science branches and information science is no different in that aspect. According to [65] “A specification of a representational vocabulary for a shared domain of discourse — definitions of classes, relations, functions, and other objects — is called an ontology”. From the point of view of computer scientists ontologies are very good candidates for sharing information about a specific domain in a formal way.

An ontology however should not be confused with a taxonomy where entities are arranged in a way that only takes the generalization and specialization properties into account. The well known online encyclopedia Wikipedia for example categorizes and sub categorizes the topics in their database in a manner where finding information is easier. The result of this categorization is a taxonomy. An ontology also defines many relations between entities, restrictions and also the way these relations are to be used.

The main components that make up an ontology are:

- Language: A way to formally define ontology concepts. Must provide enough flexibility to allow high level definitions and low level restrictions. Example: OWL
- Concept (Class): Building block of the ontology. Every entity belongs to a class. Every property is also a class. Classes allow hierarchical organization. Example : Book
- Taxonomy: This is a hierarchical organization of classes. Of course an ontology with a single class or multiple independent classes can be defined but most practical ontologies come with a detailed class hierarchy.
- Attribute (Feature): Low level descriptors defined on classes. These can be some primitive type defined on the base language or user defined types. Example: Price (decimal)
- Property (Relation): The way classes are connected. These define interclass properties. In RDF definitions this is the predicate that connects subject and object. Example author->*authorOf*->book.
- Restriction (Constraint): Defined mostly on relations. Restricts the way the relation is used. Example: *Only* defined on *authorOf* and *book* tells that the subject is suspect to using this relation only on books.
- Instance (Object): Realizations of classes. An ontology can act as a database with this ability. While a *Book* is a class, "are23*l" is an instance of that class with *title* attribute "Ontology 101" *writtenBy* "Ayşe Yazar". Of course all relations and restrictions defined on the *Book* class applies to "are23*l".

In Figure 6 a sample ontology for pizzas with different topics is given [66]. The OWL file is processed by Protégé [67] and visualized by OWLPropViz [68] plug-in.

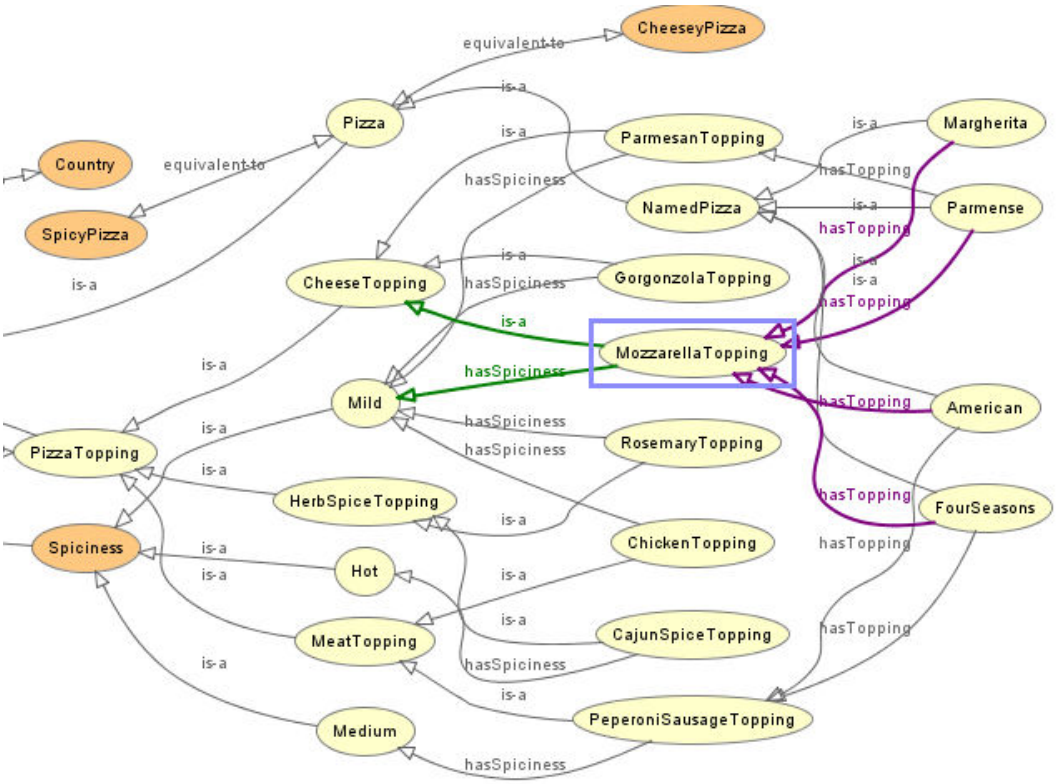

Figure 6 - A Sample Ontology

2.2.8.2 Ontology For Semantic Web

From a semantic web point of view it is evident that ontologies are powerful tools in conveying conceptual views through the web. The mechanism we have reviewed so far such as RDF and RDFS allow individuals to define concepts and some simpler relations but in order for semantic web to work we have to define a common way of describing our data in a way that the data can be related to other individuals' definitions.

For example when one information source defines "zip code" and another information source defines "postal code" we need a way to know they are equivalent. Ontologies can give such descriptions. Furthermore since an ontology contains more information than simple names it could be possible to guess that a

“zip code” is same as a “postal code” based on the fact that they are both attributes of “address” which is on the range part of “livesAt” relation.

While naming resource with URI’s is a good way identifying them we have seen that they are not unique names and thus cannot be used reliability to unambiguously define entities. However within an ontology an entity can have more meaning. “Zip” for example can be understood to define the zip code of an address in an ontology where it is part of an “address”. In another ontology that defines clothes for example we could see that “Zip” is part of “Jeans”. Now when running a query on different ontologies we could know that one refers to addresses while the other is a locking mechanism used on clothes. Such studies are already underway and research [69][70] is being conducted on the subject under the name “ontology matching” for some time now.

Another power of ontology comes from the fact that it allows inference engines to be built on top of them. Since ontologies are formal definitions of entity hierarchy, attributes, relations and restrictions it gives a solid ground on inference rules to work on. The power of the inference engine can vary depending on the needs. A very powerful engine can infer many rules while a simpler one can provide only simpler consistency checking mechanisms. Since an ontology language is independent of the applications that use it, it is possible to use the same ontology in web applications, desktop applications and embedded systems while developing the application logic in any computer programming language. So the processing power required to implement the inference engines can vary according to application platforms. So the ability to have variable powers of inference engine is a plus.

One misconception while talking about ontologies and semantic we is, it is assumed that a huge central ontology where everything is defined will be needed to have a common ground to talk about knowledge. First it should be noted that semantic web vision does not require a perfect system. [50] States “Semantic Web researchers, in contrast, accept that paradoxes and unanswerable questions are a

price that must be paid to achieve versatility". We do not need a way to represent the same data identically everywhere on the web. In fact that could seriously limit creativity and research. Instead semantic web aims to provide mechanism where knowledge can be shared in a structured manner so that machines can comprehend that knowledge as well. There will be inconsistencies, ontologies that define the same concepts in totally different ways and confusion. But that itself is no different than the way we use WWW today. Allowing for a system to let machine process huge amounts of data for us will only benefit information systems and humans in general.

2.2.8.3 Ontology Language For Web: OWL

Realizing the power of ontologies and their importance for the semantic web development groups from America and Europe started working on developing a language that had more expressive power that RDF and RDFSchema. The American group named DARPA came up with the ontology description language DARPA Agent Markup Language (DAML) [71]. As part of IST OntoKnowledge project the European group came up with OIL (Ontology Inference Layer or Ontology Interchange Language) [72]. Later the two languages were combined and named DAML-OIL which is the first ontology language for semantic web that had widespread acceptance.

In 2001 W3C started the consortium Web Ontology Working Group which was disbanded in 2004 [73] after releasing a formal W3C recommendation that describes OWL(Web Ontology Language). In 2007 W3C another group stated on a new version of which is now known as OWL 2.0 which is compatible with OWL1. The OWL2 became a W3C recommendation in 2009 [74]. OWL is largely based on the works of DAML-OIL. The language used RDF/XML syntax and defines the syntax to express classes, attributes, relations and restrictions that make up an ontology.

The whole features presented in OWL can be found in OWL documentation [74]. An overview of the more important features is given below. Examples are in RDF/XML Syntax.

- **Class:** The ability to define classes. Defines "Mary" instance to be a member of Person class and also Woman class.

```
<Person rdf:about="Mary"/>
<Woman rdf:about="Mary"/>
```

- **Class Hierarchy:** The ability to connect classes in is-a relation.

  Examples state: "Woman" is a "Person" and "Mother" is a "Woman". While this establishes the hierarchy between classes it also implies that a "Woman" is a "Person" without explicit coding. Third example shows the system that we can use "Human" instead of "Person" i.e. they are equivalent. Last Example shows that the classes "Woman" and "Man" are disjoint. This allows inference engines to do a consistency check on instances.

```
<owl:Class rdf:about="Woman">
        <rdfs:subClassOf rdf:resource="Person"/>
</owl:Class>
<owl:Class rdf:about="Mother">
        <rdfs:subClassOf rdf:resource="Woman"/>
</owl:Class>
<owl:Class rdf:about="Person">
        <owl:equivalentClass rdf:resource="Human"/>
</owl:Class>
<owl:AllDisjointClasses>
        <owl:members rdf:parseType="Collection">
        <owl:Class rdf:about="Woman"/>
        <owl:Class rdf:about="Man"/>
        </owl:members>
```

```
</owl:AllDisjointClasses>
```

- **Object Properties :** The ability to define the relations between objects. These are more general relations beyond the scope of is-a. Example illustrates how “John” can be shown to have a wife “Mary”. The “NOT” operator also can be applied by means of owl:NegativePropertyAssertion.

```
<rdf:Description rdf:about="John">
        <hasWife rdf:resource="Mary"/>
</rdf:Description>
```

- **Property Hierarchy:** Just like classes the properties can have inheritance relations. In the example it is shown that “hasWife” property is a sub property of “hasSpouse”

```
<owl:ObjectProperty rdf:about="hasWife">
        <rdfs:subPropertyOf rdf:resource="hasSpouse"/>
</owl:ObjectProperty>
```

- **Domain and Range :** The properties we defined have a subject and an object. The values a subject can have are called the “domain” while the object domain is called the “range”. In example it is shown that the “hasWife” is defined over “Man” onto “Woman”. Given a relation, its range and domain we can define an instance relation and the inference engine would then be able to determine the subject of the relation is a member of the domain.

```
<owl:ObjectProperty rdf:about="hasWife">
        <rdfs:domain rdf:resource="Man"/>
        <rdfs:range rdf:resource="Woman"/>
</owl:ObjectProperty>
```

- **Equality/Inequality:** Previously we have pointed out that different URI's in the www can point to same objects and there is no way of knowing for sure. OWL provides the mechanism to state whether two instance are the same or different explicitly. In the examples we are shown that "John" and "Bill" are different individuals while "James" and "Jim" are the same.

```
<rdf:Description rdf:about="John">
        <owl:differentFrom rdf:resource="Bill"/>
</rdf:Description>
<rdf:Description rdf:about="James">
        <owl:sameAs rdf:resource="Jim"/>
</rdf:Description>
```

- **Data Types:** As the last step OWL presents a way to define actual data in the ontology using XML syntax. In the example it is shown that John is 51 years old

```
<Person rdf:about="John">
<hasAgerdf:datatype="http://www.w3.org/2001/XMLSchema#integer">51</hasAge></Person>
```

2.3 Clustering Techniques

Human have an intrinsic tendency towards clustering so in everyday life we cluster items, event and ideas sometimes even without realizing.

For example we cluster humans;

according to their ages; child/teenager/adult/senior.

according to their origin; Hispanic, African, Asian etc..

according to their eye color; brown, hazel, blue eyed etc..

Of course a car insurance salesman would be more interested in clustering people according to the car values, or a bank manager would like to know how people can be categorized according to their income levels.

[35] defines clustering as "… the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters)." The idea has been widely used on many research areas over the years. The simplicity of the idea and its applicability to real world makes understanding and visualizing the process more easy. However clustering becomes more complicated when dealing with high feature data and clustering features are not obvious.

Data clustering tasks have some common steps [36].

2.3.1 Pattern (Data) Representation

This can be seen as preparation to clustering itself. Includes identifying input items to feed into the clustering algorithm, identifying the features of items to be used in the algorithm, probably generating new features from existing ones and in some cases predetermining the number of classes to extract from the data.

2.3.2 Pattern (Data) Proximity (Distance)

A method to measure the distance between two data items is necessary for clustering. Simple Euclidian distance is an example of such a distance functions. Many distance functions have been proposed over the years but the applicability and precision of the metric is highly dependent on the domain and the data. Some authors propose system to learn distance metrics on the given data automatically too [37][38].

Some of the well known metrics for distance calculations are given below

- Euclidean distance = d = $\sqrt{\sum_i (a_i - b_i)^2}$
- Manhattan distance = d = $\sum_i |a_i - b_i|$
- Cosine Similarity = d = $\cos^{-1} \frac{a \cdot b}{\|a\| \|b\|}$
- Hamming or Levenshtein Distance for string distance
- For sets of distances Max/Min/Average/Mean distance

In the world of web usage mining we are interested in clustering web pages, users, user behavior and association rules. So a distance metric needs to be devised to calculate distances between said items. As completely new metrics can be devised based on the research and data, one of the aforementioned metrics or a compilation of them can also be used to determine item distance.

2.3.3 Grouping

This step consists of the actual grouping of data into different clusters. In this step the chose metric in the data distance step is applied to items that are generated in data representation step. Since clustering is used in a large area of scientific study many clustering algorithms have been proposed over the years. [39] gives a good overview of commonly used algorithms.

2.3.3.1 Hierarchical Methods

Data is grouped into tree nodes where each node has a parent and siblings. The model is known as a dendrogram. Agglomerative (bottom up) approaches start

with each data as a single cluster and apply some logic to merge the nodes into larger nodes [40]. Divisive (top down) methods start with the whole data as a single cluster and adaptively divide the clusters into smaller siblings [41]. Usually number of clusters/number of depth is given to stop the process at some point but it is possible to clusterize the whole data. Hierarchical clustering allows the data to be viewed at different levels of refinement.

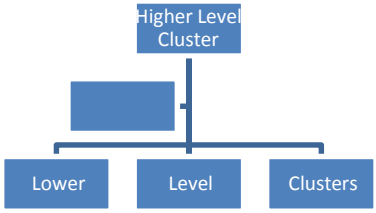

Figure 7 - Hierarchical Clustering

2.3.3.2 Relocation Methods

This group of algorithms assign some clusters in the beginning and moves (relocates) the data points among the clusters until a criteria is met i.e. no more relocation is needed.

Probabilistic methods are in this group where it is assumed that the whole data is made of smaller probabilistic distributions and data points are assigned to probabilistic mathematical representation with a probabilistic model. See [42] for sample description of Exception-Maximization method.

Very well known K-means [43][44] and k-medoids [45] algorithms are also in this group. The main difference between the two is in k-medoids one of the actual data points is assigned to be center of cluster where in k-means it is the median of the points in the cluster.

K-means is the most widely used algorithm in data clustering and various improvements/variations have been published since it was first introduced. One of

the main disadvantages of the method is that the algorithm does not decide how many clusters are present in the data. Instead the user has to tell the system the number of clusters and even the starting point of these clusters before relocation is initiated. The number and starting points of cluster greatly affects the outcome of the algorithm. [46] Proposes a method to find the natural number of clusters called the *silhouette*.

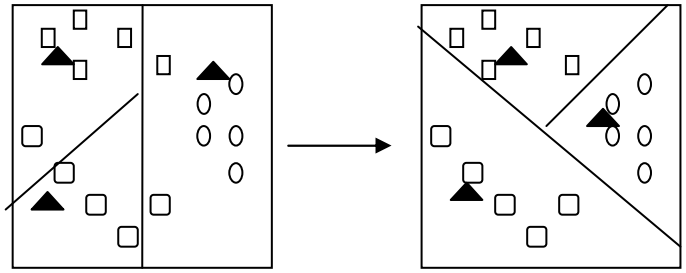

Figure 8 - K-means implementation. Relocation of centroids

2.3.3.3 Density Based Approaches

Sometimes the way data spaced can be problematic for relocation methods to find good partitions. This is usually true for clusters that are interconnected or very irregularly shaped. Some examples are seen in Figure 9.

Figure 9 – Irregular Groups of data (figure adapted from [39])

In the above cases it would be difficult for k-means to find two different group's data as both medians are very close and clusters are inside another. Density based methods however are forms of nearest-neighbor algorithms. Clusters stretch and grow to where density of points leads. If density falls below a certain value the cluster boundary is drawn there. These methods are most often applicable to spatial data such as seen in GIS applications. See [47] for GDBSCAN algorithm.

2.3.3.4 Grid Based Approaches

In these approaches the underlying problem domain is partitioned into grid structures and data points are assigned to grid space. This requires a low dimension numeric problem domain and is usually used in spatial data partitioning.

CHAPTER 3

PREVIOUS WORK

Many techniques have been employed to mine the data from server logs. [75] gives four main branches of web usage mining; however the boundaries are not always clear as most approaches utilize more than one technique in a complete system. Previous works in the area that utilize clustering techniques and benefit from semantically enriched methods are summarized below.

3.1 Clustering In Web Usage Mining

Clustering is an accepted and widely used data mining technique. Numerous research articles have been published and many algorithms have been devised to handle different types of data in different domains. The way data is partitioned, nature of the data, output of partitioning and the aim of partitioning all have important effects on the type of algorithm to be used. [76] gives a very good classification and overview of the commonly used algorithms.

Note that when talking about clustering we are referring to segmentation of raw data into different groups. Some of the articles in the field refer to association rule mining and sequential rule mining as clustering methods too. However the rules generated by such methods have a certain support and confidence value associated with them which make the result more of estimations rather than bounded classes.

In the area of web usage mining we see some different approaches in choosing what to cluster and how to cluster them. Most common way of clustering

seen is clustering the sessions. [77] proposes a way to cluster user-click stream data by defining each user session as a *path* in the website by building a graph of the web site according to the links present in each page to other pages. Clustering is applied on the paths. [78] defines each user session as a binary list of size N where N is the list of all possible pages in the web site. Where session is defined as

$$s = \{p_1, p_1, p_1, p_1, p_1, \ldots, p_N\}$$

$$\mathrm{Pj} = \begin{cases} 1, & \mathrm{Pj}\ accessed \\ 0 & else \end{cases}$$

Using s as a vector a variant of cosine similarity is defined. Also a synthetic similarity function is introduced based on the topology of the web site. Combining these similarity metrics and applying a modified CARD [79] algorithm which is a fuzzy clustering algorithm, clusters are generated.

When a new user session is introduced a vector for that session is also created and the closest cluster is chosen according to the distance metrics used in clustering. Next, a subset of the pages in the cluster is recommended based on their occurrence frequencies in that cluster. [80] applies the same ideas uses used multivariate k-means clustering instead of CARD. Moreover a possible idea of clustering URL's is suggested based on their frequencies in sessions suggested algorithm is Association Rule Hypergraph Partitioning [81]. In [82] the authors introduce the idea of using a sliding window on the current user session to limit the impact of old pageviews in the session.

Based mostly on previous work [83] proposes a latent base approach while clustering the data. Author proposes clustering URL's and assigning pageview to the corpus directly which results in a session pageview matrix. Applying single value decomposition algorithm on the matrix and cosine similarity metric. As opposed to other approaches author assigns weights on the pages according to the

visit time on them instead of assigning binary weights based on whether the page is accessed or not.

Another idea has been proposed by [84] where content mining is integrated into web usage mining. In this approach the content of the pageviews are integrated with usage profiles. In the preprocessing step of system authors apply a feature extraction step on all pages in the web site. From formal definitions

Pageview $p = <(f_1,w_1), (f_1,w_1), (f_1,w_1), (f_1,w_1),\ldots,(fn,w_n),>$

where each f_j is a feature and each w_j is the corresponding weight on the feature. Features predetermined properties that can be extracted from web pages. Feature weights are assigned by a domain expert. This step requires high domain knowledge and manual work.

In this work the authors take one more step and invert the pageview feature matrix to obtain a vector list of features where each element of the feature vector has the weight of the associated pageview. Clustering takes place on the features instead of pageviews. Authors claim that this enables to group together features most accessed by users so that a granulation on feature level can be achieved instead of page level as opposed to [85] where authors cluster the web pages themselves instead of the features. The recommendation phase of the system is a collaboration of content and usage mining. Both results are compared and the highest scoring pages are extracted as recommendations.

Unsupervised algorithms have also been proposed in the area. Self-organizing Maps (SOM) algorithm [86] has been employed by [87]. Authors have experienced that SOM generates tighter cluster compared to k-means.

A more novel approach by [88] uses artificial ant colony clustering and linear genetic programming (LGP). First the cleaned raw data is fed into ACLUSTER [89] and the result is used as input for LGP. The ant clustering is observed in real world. When ants are in a closed space and there are dead ants

and ant larvae in the space, ant will cluster the two in different groups freeing up space in the process. “The general idea is that isolated items should be picked up and dropped at some other location where more items of that type are present” [88]. This is an iterative approach and has to be cutoff at some point. See Figure 10 for the evolution of items.

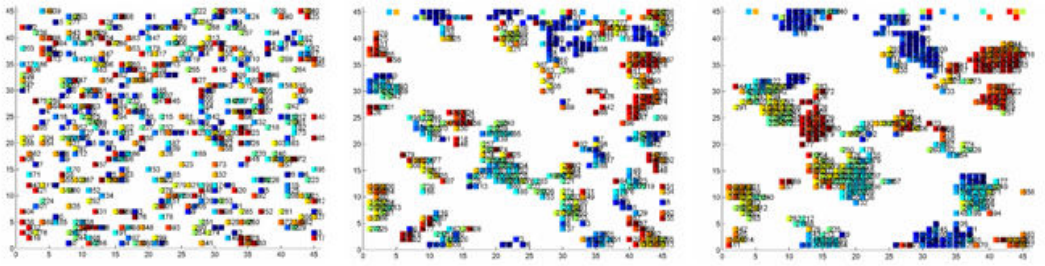

Figure 10 - Ant Clusters at t=1 t=100 and t=900 [88]

3.2 Integrating Semantics

The techniques we have seen so far for web personalization based on user web site usage data did not take semantics into account. The statistical techniques are purely mathematical and do not know or say anything about the data itself. Association rule miners are based solely on the click streams. Assigning a database ID for each item and abstracting the data itself do not have any effect on these methods. However that also shows they do not exploit the available data but work only on a fraction of it. Content integration and content based clustering methods do make use of the content by extracting feature from web pages however they cannot answer more important questions as to what makes some of the users group together on a particular resource while other user groups focus on other resource by showing the content.

None of these methods however answer the question “Why?” as they lack the semantic component in their workings and thus cannot penetrate into more

complex relations and properties of concepts that reside in the web pages. And so is the aim of semantically enriched web usage mining. However since the rise of semantic web is recent there have not been much research on combining web usage mining with ontologies. Most of the research focuses on matching ontologies[69] and automated ontology extraction from web sites and ontological concept extraction from web pages[91].

In [90] the authors present some formal descriptions and an approach that uses the power of ontologies is given. The general idea is to extract domain level objects from user sessions and create a user profile for each user by aggregating these objects according to their weight and a merge function.

A class is defined as the representation of concepts in the ontology. A class has a set of attributes (a_1, a_2, a_3,…,a_n). These attributes can be simple literals or complex objects. A merge function is defined as an operator that takes 2 attributes of the same kind and returns a combination of them.

The assumptions are that there already exists a domain level ontology for the web site. The ontology can be manually constructed or automatically generated if possible. Also all web pages are assumed to go through an information extraction phase where each ontological instance in the web page is extracted and recorded. Author also assumes the existence of merge functions defined on every attribute of objects. Merging refers to creating one aggregate object instance from multiple instances.

Given instance objects (o_1, o_2, o_3, o_4,…,o_n) of the same kind each one with attributes (a_1, a_2, a_3,…,a_m). An aggregation of first property is defined as

($<a_{11,}w_1>$,$<a_{12,}w_2>$,$<a_{13,}w_3>$,…,$<a_{1n,}w_n>$) where w_j is a weight associated with object o_j. Aggregation of the whole object set is the total aggregation of all attributes. The author believes to capture inter class relations by aggregating subjects and objects of these relations. So a separate operation on relations is not defined. As an example a simple ontology is presented from the original work in Figure 11.

The author aims to capture all ontological objects that a user visits in his session and aggregate them in order to create a user profile. Let U1 be an hypothetical user that visited two movies in one web usage session, first movie being “Spy Game 2002 Action Robert Redford Brad Pitt” and the second being “Snatch 2000 Comedy Jason Statham Benicio Del Toro” and spent 6 seconds

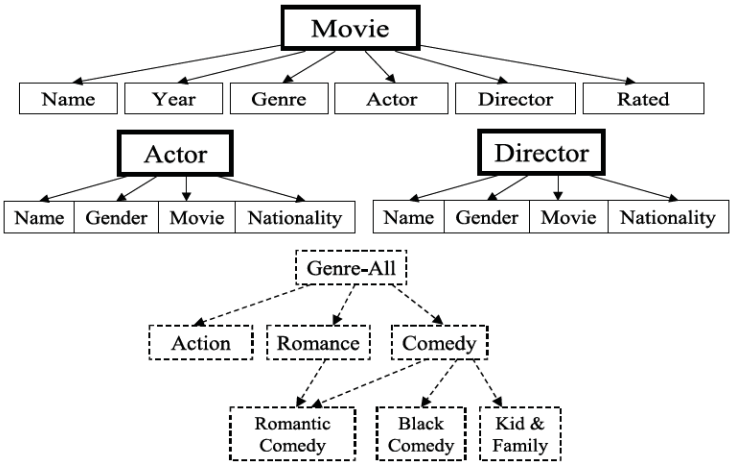

Figure 11 - A sample movie ontology

viewing the first movie while spending 4 seconds on the second. If we define genre merging as the least common, genre = Genre-All. If we define year merging as timespan, year = [2000-2002]. If we define actor merging as simple actor additions, actor = {0.6 Robert Redford, 0.6 Brad Pitt, 0.4 Jason Statham, 0.4 Benicio Del Toro}.

Given a user session as a set of domain objects extracted from pageviews as usage profile then is constructed by merging every item of the same kind together.

In this work authors cluster the sessions. Each session centroid is then calculated according to the merging procedure just described. In this way a cluster centroid and a single user session is represented in the same.

In the recommendation phase current user session is converted to a usage profile and the best matching cluster is found. Then items from this cluster are recommended to the user. Author does not go into detail as to how different usage profiles are distance compared or how the items are recommended.

Another article [92] combines apriori algorithm with item and property insertion to mine frequent patterns. The author uses travel ontology [93]. As opposed to most ontologies author uses a two level taxonomy. While the first taxonomy branches between concepts a second one branches between relations defined between concepts. For example destination->urban_area->town is a taxonomy of concepts, hasActivity->hasAdvanture->hasSafari is a taxonomy of relations. has Activity is a relation defined on destination concept.

The xPMiner algorithm proposed by author has the classical properties of apriori algorithm. First candidates of level 1 are generated. Among these items frequent items are selected and used to generate level 2 sequences and so on. A classical apriori algorithm generates k+1 the items using k the items and crossing them, xPMiner however uses object and property insertion alongside this operation. For example S_1 = <<Accommodation>, {}> can generate <<Beach>, {}> <<RuralArea>, {}> and <<UrbanArea>, {}> because they are sub classes of Accommodation. The same applies to relation taxonomy.

The most important contribution is the insertion of relation operators in the rules. This is done beginning with third step pattern generation. Rules such as <<Destination,Accommodation>, {hasAccommodation (1, 2)} are generated. This means that this rule applies if destination and accommodation are accessed in a session and also there is a hasAccommodation relation between the two.

In the recommendation phase active session is checked against all generated rules to see if the session is a prefix instantiation of them. That means a full instantiation is not required but the prefix of the rule has to be matched by the session. By finding all the objects that supersede the prefix, the objects that give better matching to the rule are selected and recommended.

A recent article [94] converts each web server log file entry into a single ontology concept. After applying SPADE algorithm on the converted log file sequential association rules are generated. However unlike the standard sequential rule miners this approach yields rules that have ontological objects as antecedent and consequent. In the recommendation phase User session is also converted to a sequence of ontology concepts and according to generated association rules, recommendations are generated. By reconstructing the page file lists from ontological objects recommended pages are generated. The approach is illustrated in Figure 12 (from original work).

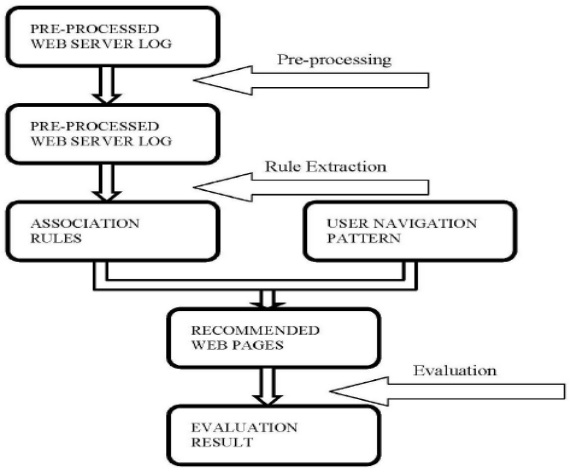

Figure 12 - General Framework [94]

CHAPTER 4

ARCHITECTURE AND IMPLEMENTATION

In this chapter the design and implementation of a recommender system is detailed. The methods used in various parts of the system as well as algorithmic details are presented.

4.1 General Architecture

The general architecture of the system has the basics of any other web usage mining system (See Figure 2). There is the data acquisition part where data is obtained. There is a data cleaning part where log files are cleaned of irrelevant information. There is an offline mining part and at last the recommendation phase which is the online part of the system.

In addition to classical systems there are some steps where we integrate domain knowledge into the system. (Figure 13). At the end of the data cleaning part we convert every session which are ordered page views, into ordered domain ontology objects. We applied a clustering algorithm on these converted sessions extract user groups as a combination of objects. At the recommendation phase we convert active user session into a sequence of objects too and find distance to cluster medians. These stages need a well defined distance metric between objects.

4.2 Steps And Methods

4.3 Preprocessing

In this step we prepare the data for mining. See Figure 13.

4.3.1 Data Acquisition

An anonymous online book retailer has a web page where each book is categorized and labeled. They supplied us with their server logs that keep track of users' interaction with the web site. We acquired log files of the site from 22.01.2008 to 30.01.2008 spanning a nine day period where each day log is in a separate file. The format of the log file is given below. (This dataset is used in [94])

```
date time s-sitename s-computername s-ip cs-method cs-uri-stem cs-
uri-query s-port cs-username c-ip cs-version cs(User-Agent)
cs(Cookie) cs(Referer) cs-host sc-status sc-substatus sc-win32-
status sc-bytes cs-bytes time-taken
```

Most fields are self explanatory. The fields we used were

- `date time` : the access time to the resource
- `cs-uri-stem` : The webpage visited. Note that pages in this web site are dynamic. This portion gives only the stem part of the actual web address.
- `cs-uri-query` : This is the part which determines which dynamic content is queried.
- `c-ip` : IP of the user trying to access the resource.

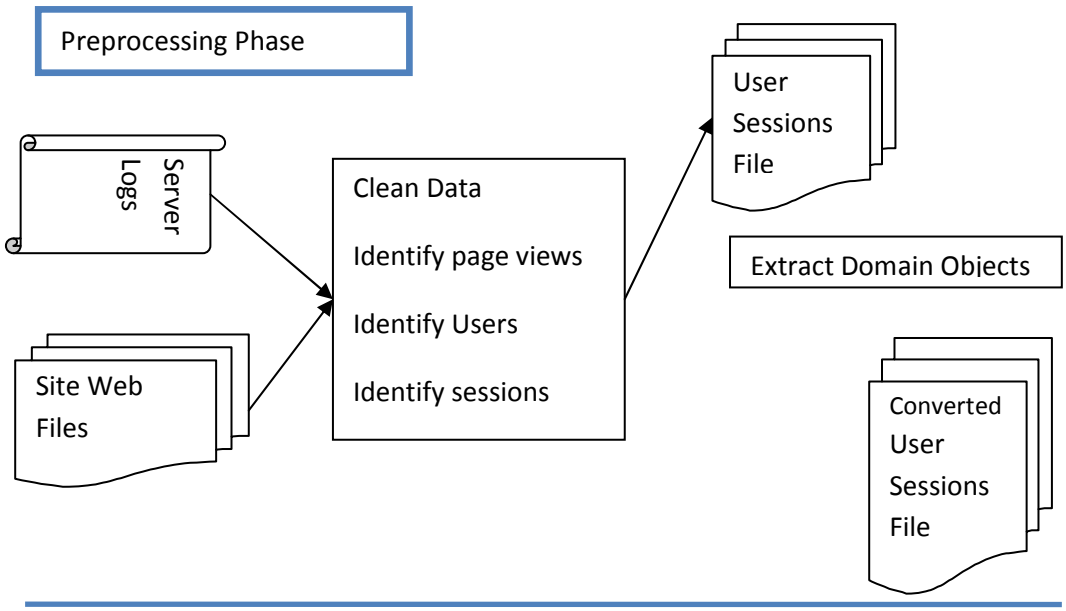

Figure 13 - Data Preperation

Two example log lines are given below.

```
2008-01-22 05:00:18 W3SVC519 SINGLE26 65.182.101.197 GET
/urun.aspx productID=9242 80 - 66.249.73.38 HTTP/1.1
Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.google.com/bot
.html) - - www.anonymousbookstore.com.tr 200 0 0 36305 297 546

2008-01-22 05:00:24 W3SVC519 SINGLE26 65.182.101.197 GET
/addtobasket.aspx productid=10300 443 - 66.249.73.38 HTTP/1.1
Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.google.com/bot
.html) - - www. anonymousbookstore.com.tr 500 0 0 4270 255 15
```

The extracted information from these two lines is

```
2008-01-22 05:00:18 66.249.73.38 urun.aspx productID=9242
2008-01-22 05:00:24 66.249.73.38 addtobasket.aspx productid=10300
```

4.3.2 Data Cleaning/Pageview Identification

The log format of this particular web page in accordance with the web page structure is a little different from most logs. There are only a few real pages in the web site. Most of the site content builds upon a few dynamic pages.

`urun.aspx`: Opens a detail page about a product.

`addtobasket.aspx`: Adds a certain product to shopping basket.

`search.aspx`: Deploys a simple search on the web site.

`AdvSearch.aspx`: Deploys an advance search on the web site.

`reyon.aspx`: Opens a category first page where items in that category are listed page by page.

The rest of the pages are pictures, style sheet documents or robot access document. We clean the log file of all these non content details.

The search pages are not of interest in this study so we discard them.

The reyon.aspx page gives a long list books in this category however the page content changes frequently based on the books recently published and it is very hard to process and extract objects from it. However the page links themselves could be of use. For the time being they are discarded.

Also discarding the fields that will not be used in mining, we are left with a simpler format. We have

the timestamp value: This value is converted to seconds passed since 00.00.0000 for easier processing.

IP address: Will be stored as a string

Page address: Since the stem part is now only limited to addtobasket.aspx and urun.aspx we will discard the stem and only hold product id from the query part.

Table 3 - Cleaned Log Format

Product_id	IP	Timestamp
7552	66.249.73.38	63336574808
9242	66.249.73.38	63336574818
10300	66.249.73.38	63336574824

At the end of the cleaning steps we merge all results from 9 log files corresponding to 9 days into a single converted file. Then the log file is sorted first according to IP numbers, then according to the timestamp values. 105879 lines of accesses have been parsed into this file. Format of the file is given below.

Table 4 - Cleaned and Sorted Pageviews

128.165.192.163	8533	63337315508
128.86.159.125	2371	63336790280
128.86.159.125	5965	63336790663
128.86.174.5	5199	63337218647
128.86.174.5	5199	63337218676
129.206.196.193	11082	63337021681
129.206.196.193	11094	63337029581
129.206.196.193	11094	63337029586

4.3.3 User Identification

The book store web site we are working on does not require authentication for browsing products so there is no certain way of identifying users. However inspection of the log file and example implementation shows that we can assume every IP identifies a unique user. This is probably due to Turkish IP infrastructure

where each internet user is assigned one unique IP at least until the connection reset.

4.3.4 Session Identification

The total of all page views by a single IP is called a session. In Table 4 we can see 4 such groups assigned to IP's

```
128.165.192.163
128.86.159.125
128.86.174.5
129.206.196.193
```

We have identified 24401 such sessions at this point of parsing the log file.

We have implemented two mechanisms to determine session breaks. First one is the time barrier. If a certain session exceeds a predetermined value of seconds, the session is broken. The new session is assigned another session id. Second barrier is the session length. If a certain number of page views is exceeded a new session is started and assigned a new id. Example format is shown in Table 5. Default time barrier used is 1800 seconds and session length is 20 page views.

After this step realizing that there are a large number of single page views we introduced a min page view count. Any session with less than min_page_view is discarded. Default value for min_page_view is 3. After applying the pruning and session breaking algorithms 5644 sessions were left to be used in our experiments.

Table 5 - First pass Session File

20	5576,2367
21	6053,8688,9952
22	5217
23	3652
24	8086,10055,9216,9368
25	9027,9320
26	8863,2710,8567

4.3.5 Mapping To Domain Objects

In this step we take each product_id in the cleaned log file and map them to domain objects defined in the ontology for this web site. Creation and details of the ontology are given in the next section.

For example a product ID of 1458 may map to book132 and author34 in the domain ontology. Each webpage is processed to extract the domain objects. Resultant file is of form

User_id – Ordered list of domain objects

1- (o1)-(p1,j1,o2)

2- (o1,p2)-(k2)

…

Also a reverse file is generated at this step. This file contains object-page relations. This file holds with whether an object is seen in a web page.

The format is

	p1	p2	p3	…
ObjectX	1	0	1	…
ObjectY	0	1	1	…

Although this is a more generic method and should cover the base for processing different data from different domains, the data that we use for our experiments limits us to using only “book” objects. As mentioned before “anonymous bookstore data” has product identification where each ID is assigned to a single book. So at each pageview we were restricted to extract only a single object that is the book assigned to the page id.

So for the purposes of this research we had a file format as

16-5286, 4835, 4835
21-6053, 8688, 9952
24-8086, 10055, 9216, 9368
26-8863, 2710, 8567
.
.

where the first number is the session_ID and the proceeding numbers correspond to a book each. Each book data has been discovered in a previous research by [94] before. One example would be

Product_id = 2679
Book_Name=Olumlu Yaşamanın Gücü
Book_Author=Norman Vincent Peale
ISBN = 975 322 020 0
Description=İnsan sonsuz bir güç odağıdır. Kişinin kendi güçsüzlüğüne ve yapamayacaklarına inanması bu güce set koyar. Olumluya inanmak bu gücü kullanabileceğimiz alanı genişletmektedir. Sağduyu, yani içimizdeki ses bu sonsuzluk içinde bizim için en doğru yönü fısıld
Page_Count =252

Price = 9500000

Publish_Date =2003

Category = Psikoloji

Publisher = Tüdav Yayınları

9544 books have been identified and stored in a database.

4.4 Ontology Creation

The book web site in this study is fairly simple in terms of data diversity. It only sells books so we did not need a complex ontology structure. The "Kitap ontology" used in [94] is going to be used. Protégé is used as the ontology creating tool. It uses OWL as the ontology description language.

The concepts defined are

Kitap: Which is main topic of interest in the web site. (Figure 14)

KitapFiyat: Holds he value for the price of the book. (Figure 15)

Kategori: Hold the taxonomical value. This is the genre of the book. This class highly categorizes and goes down 4 levels (Figure 16).

Yazar: Authors of the books (Figure 17).

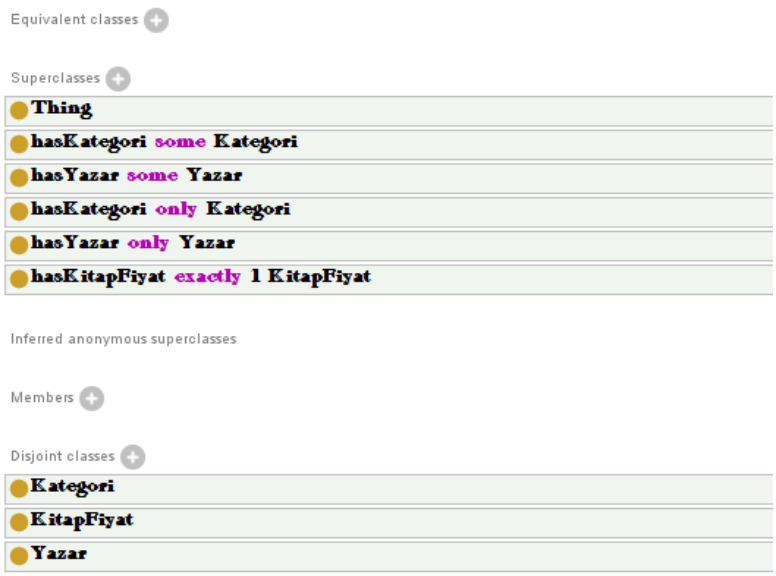

Figure 14 - Kitap Class

Figure 15 - KitapFiyat Class

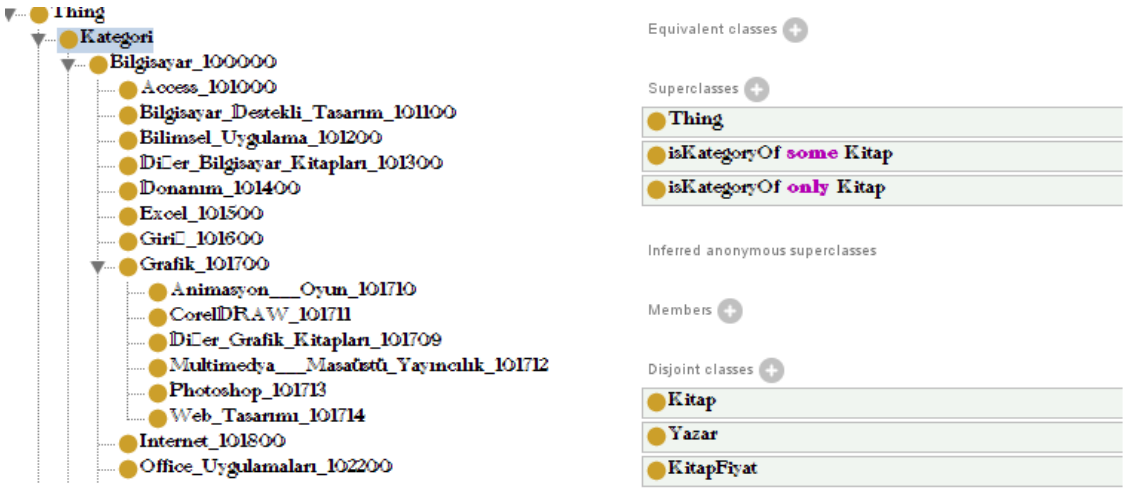

Figure 16 - Kategori Class

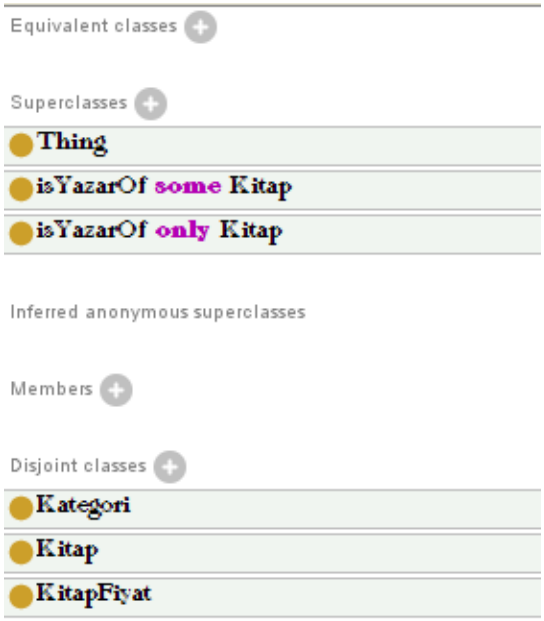

Figure 17 - Yazar Class

The relations defined on these classes are

HasDefualtCategory: Every book is assigned a default category.

Domain: Book

Range: Kategori

HasKategori: Category of the book.

Domain: Book

Range: Kategori

Inverse: isKategoriOf

hasKitapFiyat: Price of the book.

Domain: Book

Range: KitapFiyat

Inverse: isKitapFiyatOf

hasYazar: Defines the authors of the books

Domain: Book

Range: Yazar

Inverse: isYazarOf

A graphical representation of the relations between the classes is given in Figure 18.

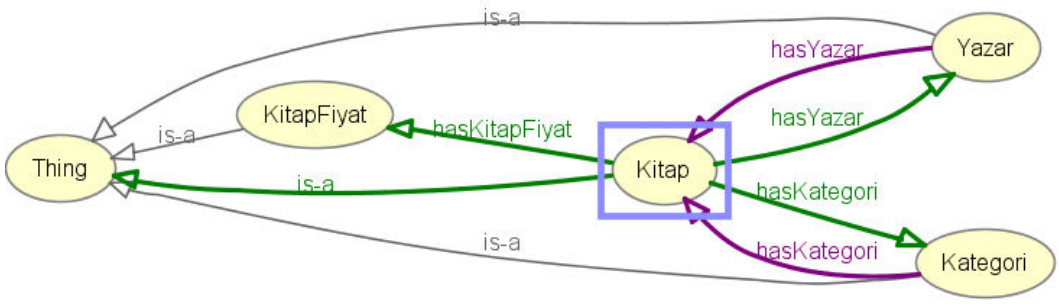

Figure 18 - Overview of Book Ontology

The attributes defined on the classes are

For *Kitap*

Aciklama (string): Description of the book.

AltAciklama (string): More detail on the book.

Ebat (string): Dimensions of the book

ISBN (string): ISBN number of the book

SayfaSayisi (int): Page count

Spot (string): Description shown on the first look page.

UrunSira (int): Number of the book.

UstAciklama (string): Any more description if available.

YayinYili (int): Publish year

For *KitapFiyat*

Fiyat1 (float): List price

Fiyat2 (float): Discount price

Indirim (int): Discount percent

*KDV(int):*Taxation percent

*ParaBirimi(string):*Currency

In this step each web page is processed and the values corresponding to each field and class in the ontology is filled accordingly. They are called instances in the ontology. An example view of a book is presented in Figure 19.

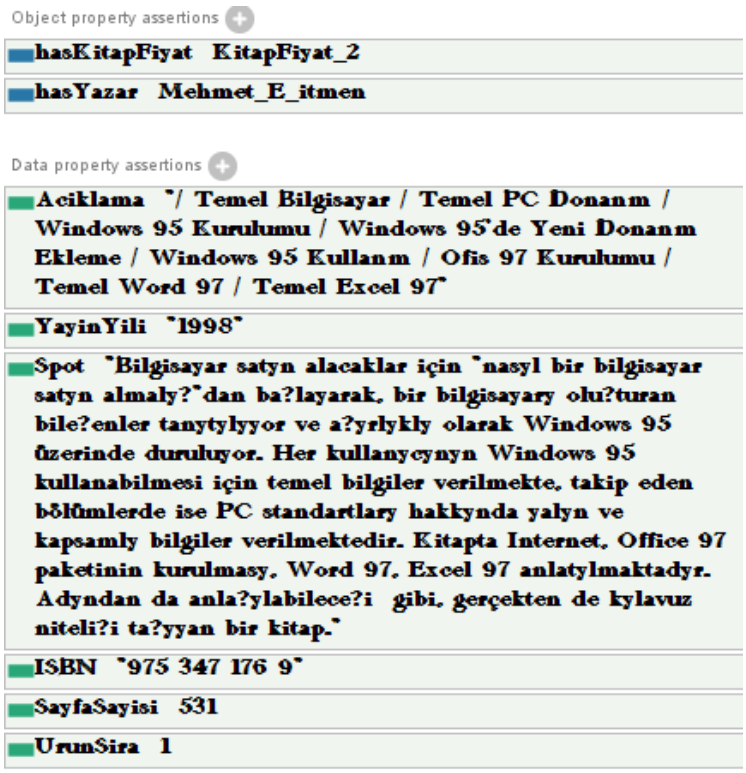

Figure 19 - A Kitap Instance

The RDF description for each book has been entered into a database for easier access. An example is given in APPENDIX A – Sample Book RDF.

4.5 Mining The Data

We use k-means data clustering method in this step. The clustering approaches we have seen so far either

- Cluster URLs visited by users

- Cluster Sessions according to page_id s
- Cluster the extracted features/domain objects directly
- Cluster domain objects after merging them, thus losing temporal relations.

In this work we introduce a new approach to using clustering for web usage mining. We cluster sessions while sessions are represented as "an ordered list of set of ontological items".

The cleaned and preprocessed log file we have at the moment has the following form.

S1 = Objects2-3-4 → objects1-3-5-8

S2 = Objects8-15-17→objects1-5-12→objects3-9-10-11

where each S is a unique session. Objects are domain mapped ontology instances grouped together in page views. Also the sequence of the visited page views is preserved.

In order to cluster these sessions we need three methods.

1- A distance metric between domain objects and a distance function to compare them.
2- A way to compare 2 sequences keeping the sequential data relevant to comparison and return a distance.
3- We need to integrate the set structure of the sequence elements into the algorithm since our sequences are not simple consecutive items but a sequence of sets of items.

For the purposes of this research where every object is a book sessions are of form

S1 = Book1->Book2->Book3….

S2 = Book1->Book3->….

The idea however will still apply because cluster means will have multiple book instances in each set.

The attributes of book objects are

AuthorName(string)

Instead of the string comparison algorithm we compare equality and return 0 distance if equal and 0.3 distance if different. We do not use Levenshtein distance because it is highly unlikely that groups of people get interested in authors with similar names. Such as user who like “Hakan” also like “Akan”

BookName(string)

We use string comparison.

Category(string)

This is a domain object and is considered separately.

Description(string)

This attribute was discarded because there is not any simple way to actually compare two different paragraphs of text and get a good distance measure. Levenshtein distance only compares letters. For comparing data such as descriptions we need context based string comparison.

Publisher(string)

Again the equal/unequal logic is used

PageCount(int)

The difference between page numbers is divided by 1000 to get a distance measure.

Price(double)

The difference between prices is divided by 10000000 (Old Lira) to get a distance value.

PublishDate(int)

The difference between publish dates is divided by 10.

After summing up the results and dividing by 7 to get the average value distance is returned. Similarity is 1/distance and if distance is 0 similarity is given as 20. This value is the result of empirical approximation.

In order to compare how different properties of objects affect the outcome of clusters we have also experimented with taking only the category into distance function and disregarding the rest of the features. The empirical results are shown in the results section.

4.5.1 Distance Between Ontological Objects

Simple similarity measures such as Euclidian distance cannot be applied on ontological objects since the distance functions assume a numerical data. However ontological objects are complex and have different types of attributes. [95][96] details some approaches to measure the distance between such objects. We will employ some of the ideas in these articles to define the similarity between our objects.

The distance between two objects can be defined as a function of the distance between their attributes and their location in the ontological tree.

Assume object as a two tuple

$$O = (A,L)$$

where A is a set of attributes (a1,a2,a3,…,an) and
L is a location representation in the taxonomy of the ontology. Any graph representation defined on a tree can be sufficient.

Then the distance between O1 and O2, DIST(O1,O2), can be defined as the weighted sum of distances of object attributes and tree locations.

DIST(O1,O2) = DistA(A1,A2) x W1 + DistL(L1,L2) x W2

where

DistA is a function that returns the distance between two sets of attributes,

DistL is a function that returns the distance between two locations in a tree,

W1 and W2 are assigned weights to the distance functions.

For the DistL we use the routing tale approach from [96]. For the DisA function we employ the attribute distance definition of [95]. We only consider attributes with simple types, either numerical or string.

The only attribute that is a domain object on its own is the Category property of the book in our ontology. So the string comparison algorithm "LevenshteinDistance" is used to compare the names of categories since that is the only attribute of the category.

So DistA for this particular domain object becomes the normalized Levenshtein distance, which is

LevenshteinDistance(string1,string2)/ Max(Length(string1),Length(string2))

The division is for normalizing the distance values.

For example if we compare "Ruya yorumlari" category with "Populer bilim" we first compare with Levenshtein the names. That comes up with 13. We divide this number with 14 to get a normalized value. This gives 0.93.

Also we see that both these categories are on 2nd level in the tree. That gives 2 level distance between them. Since there are a maximum of 5 levels and maximum distance between nodes is 8 we get 0.25 from DistL.

0.93x0.5+0.25x0.5 = 0.59 becomes the distance between category elements.

4.5.2 Comparing Sequences

Since we do not want to lose the temporal relation between page accesses we need an algorithm to compare two sequences and return a distance. Needleman–Wunsch [97] algorithm is a well known algorithm for DNA sequence matching. The algorithm also has a gap penalty mechanism that we will use to select the best matching element in a set.

This algorithm compares corresponding items in two sequences. According to the distance between items decides to either continue on both pairs or introduce a gap to one of the pairs. Algorithm aims to find the matching configuration between two sequences and computes a score. The algorithm is an example of dynamic programming.

In the initialization step a matrix is created and first row/column is filled.

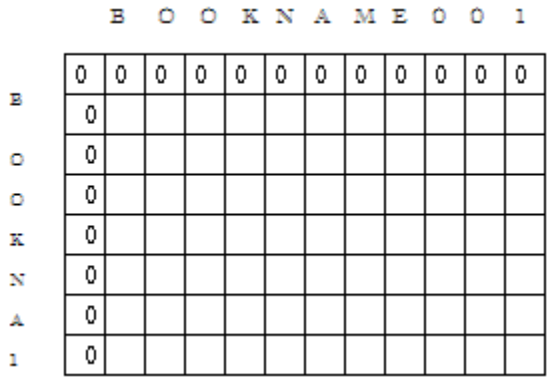

Figure 20 - Needleman–Wunsch Initialization

Then according to the pseudo code below the table is filled where A is the first sequence and B is the second sequence, d is the gap penalty and S is the similarity function defined between items.

for i=0 to length(A)

```
  F(i,0) ← d*i
for j=0 to length(B)
  F(0,j) ← d*j
for i=1 to length(A)
  for j = 1 to length(B)
  {
    Choice1 ← F(i-1,j-1) + S(A(i), B(j))
    Choice2 ← F(i-1, j) + d
    Choice3 ← F(i, j-1) + d
    F(i,j) ← max(Choice1, Choice2, Choice3)
  }
```

Resultant matrix is

		B	O	O	K	N	A	M	E	O	O	1
B	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	1	1
O	0	1	1	1	1	1	1	1	2	2	2	2
O	0	1	2	2	2	2	2	2	2	2	2	3
K	0	1	2	2	3	3	3	3	3	3	3	3
N	0	1	2	2	3	3	3	4	4	4	4	4
A	0	1	2	2	3	3	3	4	4	5	5	5
1	0	1	2	3	3	3	3	4	5	5	5	**6**

Figure 21 - Needleman–Wunsch Result

At this point the algorithm can trace back to find the best matching sequences. However we only need the value in the last score so we do not take those steps.

4.5.3 Integrating The Sets Of Objects Into Needleman–Wunsch

The idea in the Needleman–Wunsch assumes single items in the sequence steps. However in our structure we have multiple objects in each step.

S1 = Objects2-3-4 → objects1-3-5-8

S2 = Objects8-15-17→objects1-5-12→objects3-9-10-11

We can define distance of (2,3,4) and (8,15,17) to be the maximum/average/minimum distance of all items in the set. A gap penalty can be introduced to cutoff comparisons at certain level, thus achieving a pruning effect.

In our work we used average similarity. Different gap penalties have been tried but they resulted in uneven clusters. Best clusters were obtained with a gap penalty of 0.

4.5.4 Finding Cluster Means

In order to find the medians in the cluster we need a way to aggregate the objects in the sessions. Let X be a cluster of sessions.

Cluster X

	Column_1	column_2	column_3
S1 = objects1-2-5	→ objects1-2-3		
S2 = objects2-3-5	→ objects4-5-8	→ objects1-2-3	
S3 = objects1 -2	→ objects1-3		

Now we can parse each column for the frequency of objects in that column to extract the dominant objects. We can use a minsup value determine what to discard.

For example the first column has

object1 in 2 sessions out of 3 %66

object2 in 2 sessions out of 3 %66

object3 in 1 session out of 3 %33

object5 in 1 session out of 3 %33

Setting the threshold to 0.50 would contain only object1 and object2. We can assign weight to each object based on their frequency.

We can introduce a second minsup value to discard columns where very little information is present. In the given example column3 could be discarded.

The resulting cluster mean for above example with threshold 0.25 would be

M = objects1-2,objects1-3

In our work we used the minFrequency value to discard infrequent items. Unlike the example shown above the frequency limit was set to around 0.001 because there is a large number of items in each column.

Also we used a maximumItem count to limit the number of books each set could contain so that the mean would not get too large as to lose its meaning.

4.6 Recommendation Phase

This is the online component of the system. When a new user enters the web site and begins navigating the pages a session is recorded based on the visited pages.

Let's assume the user has accessed:

a.html->b.html->e.html->a.html->c.html->h.html

Considering a typical web usage scenario the user's last few page accesses determine the future path of accesses. That is why we use a sliding window of size k over the active session. Assume k =3; Active session becomes

a.html->c.html->h.html

Using the same techniques while converting offline user sessions into object sequences we convert the above page view sequence into

object1-2-4→object1-2→object1-4-5

The cluster with the smallest distance is found. Note that the cluster medians we found in the clustering step are in the same format as user sessions. So finding the distance between a cluster and a session is no different than finding the distance between two sessions.

Now we go through all sessions in this cluster and find the best matching *n* sessions to the active session. Using the reverse object file generated before we turn the objects in these sessions into html equivalents and recommend a page accordingly. The number of books in the most similar session varies based on the best matching session itself. In order to limit the number of book recommendations we are enumerating the books in the most similar session according to their frequency of appearance in the global session file. Top 5 books are then recommended.

CHAPTER 5

EXPERIMENTS AND RESULTS

The methods employed in the implementation and some of the intermediate results have been presented in previous chapter. In the chapter we summarize the intermediate results and explain the experiments we have conducted to measure the effectiveness of the proposed system.

5.1 Parameters and Methodology

The number of books in the database : 9544

Number of sessions found after log parsing: 5644

For cross validation purposes we divided the input sessions into 3 sets. First sets uses the first 3500 sessions of 5644 total sessions for clustering and the rest for recommendation trials. The second set uses the middle 3500 for clustering and the last set uses the last 3500 of the sessions for clustering. The rest are used for recommendation.

The parameters for generating the clusters are

minFrequency	denoted as X
maxItemsInMean	denoted as Y
k value for K-means	denoted as K

Cluster start centroids are chosen randomly from the sessions.

For evaluation the remaining 2144 sessions were used. The evaluation process is as follows.

Take a session if it has at least 6 books in it. (There were 1018 such sessions)

S = A-B-C-D-E-F-G-X-Y-Z

Take out the last 3 books that have been accessed

S1 = A-B-C-D-E-F-G

For each cluster that have been generated previously calculate the distance

Distance(ClusterMeanCx,S1)

Get the cluster (C) with the closest mean to S1.

For each session in C calculate the distance

Distance(SessionX,S1)

Get 3 Sessions that have the least distance to S1 (Sx,Sy,Sz)

Order the books in Sx,Sy,and Sz according their frequency in the global session set and take the first 5 and use them as recommendations. Globally most frequent means, the books that are most frequently seen in the session file.

The first candidate is Sx since it is the best matching session to S1.

If Sx contains book X we increment counter 1 2 3

If Sx contains book Y we increment counter 2 3

If Sx contains book Z we increment counter 3

The idea is that even if our recommendation is not the same as the book in immediate sequence which is X, still the session continues with Y and Z so they are partial successes.

We apply the same method to Sy and Sz as next set of recommendations.

This process gives us a 3x3 matrix where the first row belongs to Sx and is expected to have the least accuracy and last row belongs to Sz and has the best accuracy. However first row achieves its accuracy with lower number of recommendations.

5.2 Experiments

First the effect of the parameter K is observed by holding X and Y the same and varying K. We have tried K = 5,10,20,30,50,75,100 while setting X = 0.001 and Y = 3.The generated clusters for the first parameter set for the first dataset are as follows.

ClusterSet1 time 00:24:23.2968750
X = 0.001 Y =3 K = 5
cluster 1 node count = 203
cluster 2 node count = 622
cluster 3 node count = 35
cluster 4 node count = 2048
cluster 5 node count = 592
intraDistance = 0.035659
interDistance = 0.113374

The intraDistance value is the average distance of all nodes in a cluster to the cluster median. This value shows how close every node in the cluster is to each other. Lower values mean more compact clusters.

The interDistance value is the sum of distances of all clusters in the set to each other. This value shows how close each cluster is to each other in the set. Higher values mean cluster are far apart.

Best clusters are formed with low intraDistance and high interDistance values. The same results were obtained using different K values for all 3 validation sets. Table **6** shows the results in tabular form. Values for set3 could not be computed due to time constraints.

Table 6 – Varying cluster count. Distance values

cluster count	Set1		Set2		Set3		average	
	intra	inter	intra	inter	intra	inter	intra	inter
5	0.0356	0.1133	0.0360	0.0637	0.0270	0.06376	0.032944	0.080303
10	0.0349	0.7569	0.0326	0.1514	0.0244	0.147109	0.030686	0.351843
20	0.0308	0.8381	0.0302	0.3135	0.0237	0.308975	0.028302	0.486887
30	0.0318	1.7260	0.029	0.4895	0.0243	0.647337	0.028502	0.95429
50	0.0301	2.8127	0.0269	0.8610	0.0474	3.662882	0.03487	2.445553
75	0.0253	4.2940	0.0250	1.306	-	-	0.025218	2.800352
100	0.0241	7.3302	0.0241	1.7345	-	-	0.024133	4.532394

The distribution of cluster distances is shown in Figure 22 and Figure 23. From these results we see that high cluster counts give lower intra cluster distance and high intra cluster distance. We have concluded that a cluster count of 10 gives both low intra cluster values and relatively high inter cluster values. So in the following experiments a cluster count of 10 will be used.

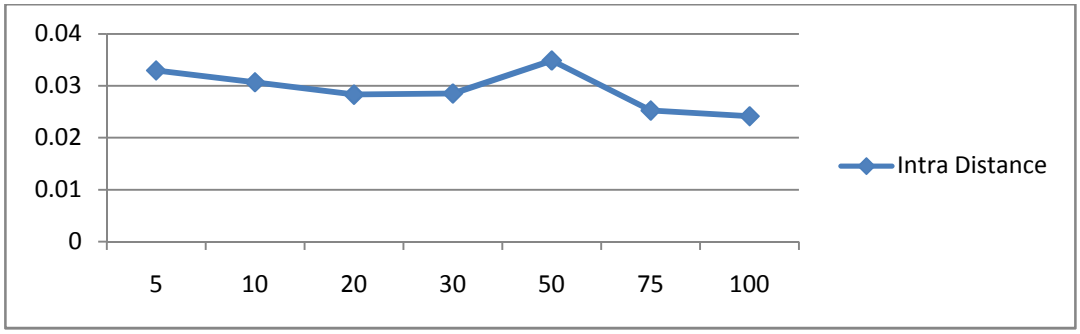

Figure 22 – Varying K – Intra cluster distance

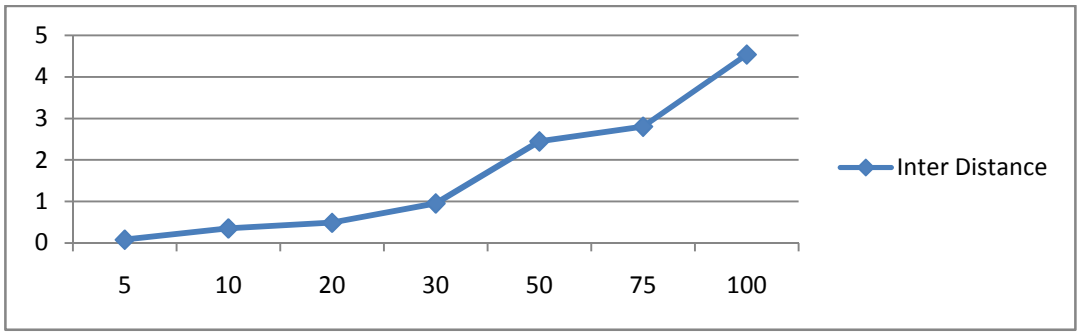

Figure 23 – Varying K – Inter cluster distance

Next we will observe the effect of parameter X(minFrequency) on the clusters. We will set K =10 and Y =3 while changing X according to the formula

X = 0.001 * EXP(i) where I =0 to 8

Again 3 cross validation sets were used and an average was calculated. Results in tabular form are shown in Table 7, and the average is shown graphically in Figure 24 and Figure 25.

Table 7 – Varying minFrequency. Distance values

minFrequency	Set1		Set2		Set3		average	
	intra	inter	intra	inter	intra	inter	intra	inter
0.001	0.0332	0.7257	0.0355	0.1501	0.03324	0.7257	0.03400	0.5338
0.002	0.0345	0.7247	0.0332	0.1517	0.03453	0.7247	0.03410	0.5337
0.007	0.0498	0.9842	0.0321	0.1525	0.04981	0.9842	0.04392	0.7070
0.020	0.1026	2.1036	0.1026	2.1036	0.10262	2.1036	0.10262	2.1036
0.054	0.0526	1.5061	0.0843	0.4372	0.05266	1.5061	0.06323	1.1498
0.148	0.0854	1.7755	0.0708	0.7200	0.08543	1.7755	0.08056	1.4237
0.403	0.0389	0.6045	0.0863	0.4643	0.03899	0.6045	0.05479	0.5578
1	0	0	0.0103	0	0	0	0.00344	0

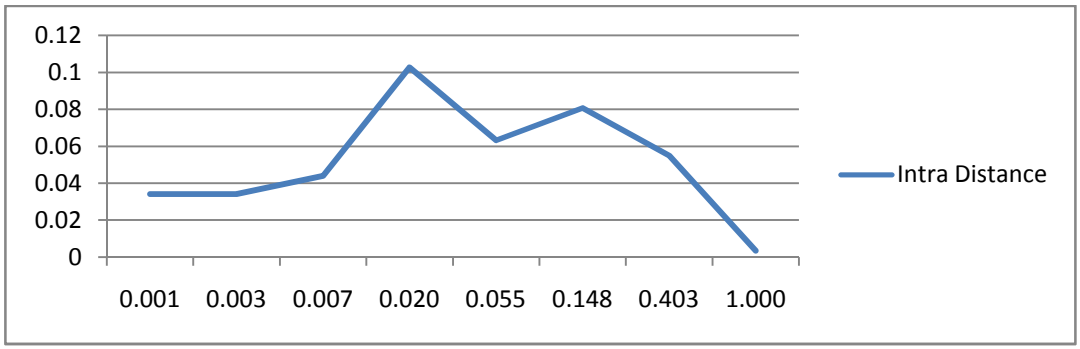

Figure 24 – Varying minFrequency – Intra cluster distance

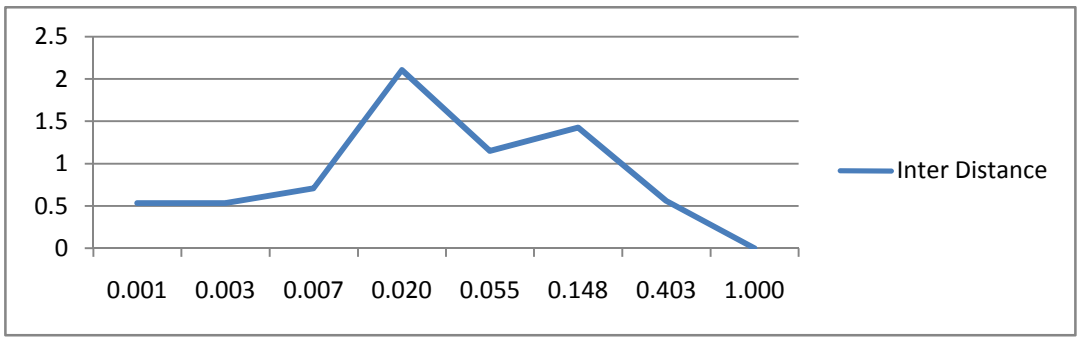

Figure 25 – Varying minFrequency – Inter cluster distance

We see that increasing the minFrequency limit increases both distance metrics. However as minFrequency goes to 1 cluster generation becomes impossible since no book has a frequency of 100% so means cannot be calculated. From the graphs we conclude that choosing an X of 0.001-0.003 is optimal so in our next experiments we will use X=0.001.

Lastly we will examine the effect of parameter Y(maxItems) to cluster generations. Again results are presented in tabular format
Table **8** and average results are shown in graphs Figure 26 and Figure 27.

Table 8 – Varying maxItems. Distance values

maxItems	Set1		Set2		Set3		average	
	intra	inter	intra	inter	intra	inter	intra	inter
1	0.0230	0.5532	0.0302	0.1244	0.02243	0.1498	0.02524	0.2758
2	0.0279	0.6206	0.0308	0.1329	0.02258	0.1494	0.02714	0.3010
3	0.0332	0.7257	0.0344	0.1372	0.03375	0.3746	0.03380	0.4125
4	0.0362	0.7369	0.0369	0.1344	0.13269	0.9055	0.06864	0.5923
5	0.0374	0.5214	0.0387	0.1357	0.05003	0.4428	0.04208	0.3667
6	0.0401	0.5240	0.0406	0.1348	0.38510	1.4343	0.15530	0.6977
7	0.0408	0.5272	0.0402	0.1357	0.02	0.192	0.03369	0.285
8	0.0418	0.5333	0.0417	0.1345	0.01	0	0.03122	0.2226

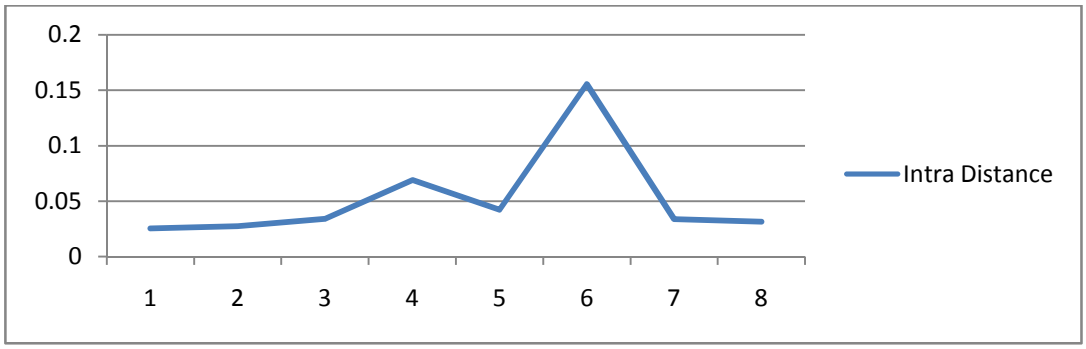

Figure 26 – Varying maxItems – Intra cluster distance

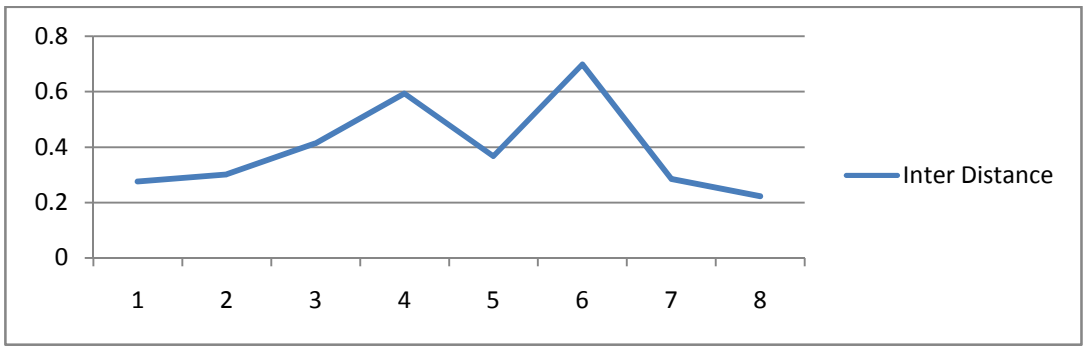

Figure 27 – Varying maxItems – Inter cluster distance

From the graphs we see that increasing maxItems increases both distances. However increasing this parameters increases execution times linearly too. So the value of 3 was decided as best since it both gives a reasonable execution time and low intra-cluster distances.

It is decided that setting X=0.001 Y=3 and K = 10 gives the best clusters so we will generate recommendation results for this parameter set. We will then compare the results to another set of results where semantic information is not used, sequence structure and temporal information is not used and a random recommender`s recommendation results.

The results for the first experiment where X = 0.001 Y = 3 and K=10 is as follows.

time = 03:08:42.1250000

checkedSessionsCount = 1018

reccomendCount1 = 4997 successCount1 = 44 56 66

reccomendCount2 = 9997 successCount2 = 282 407 512

reccomendCount3 = 15010 successCount3 = 401 572 693

Checked session count gives the number of session with more than 6 books in it. It is 1018 and is the same for all cluster sets.

RecommendCount1 gives data related to best matching session, number of books recommended by it and successful recommendations. For example a total of 4997 books were recommended by best matching sessions. 4997/1018 ~= 5 so on average 5 books were recommended to each active session, which is the expected value.

Out of the 1018 sessions 44 of them received exactly the same book recommendation as they actually accessed. We will now calculate recall and precision values for our system.

Recall is calculated as the ratio of successful recommendations to number of all possible successful recommendations. In this case if 44 of the 1018 session

received successful recommendations recall is 44/1018. It would be possible to increase recall to 100% by recommending all the books but this would significantly reduce precision.

44 /1018 = %4.32

56 /(2*1018) = % 2.75

66/(3*1018) = % 2.16

Going back to the running example

Assume we are checking S = ABCDEFGXYZ

We use S1=ABCDEFG as the active session.

4.32% of the 1018 correct answers were given (X) as part of the recommended 5 books.

2.75% of the 1018*2 correct answers were given (X or Y) as part of the recommended 5 books.

2.16% of the 1018*3 sessions answers were given (X or Y or Z) as part of the recommended 5 books.

Precision is the ratio of successful recommendations to all recommendations made. This shows the accuracy of recommendations. If too many books are recommended precision drops. In our case maximum precision that can be achieved is 20% for finding the first best match, since we are recommending 5 books each time and only 1 can be the correct answer.

Precision is calculated as

44/4997 = 0.88%

56/4997 = 1.12%

66/4997 = 1.32%

ReccomendCount2 counts both the best matching and the second best matching session recommendations.

9997/1018 = 10 books were recommended on average. Recall values are

282 /1018 =%27.701 contained X

407 /1018*2 =%19.99 contained X or Y

512 /1018*3 =%16.76 contained X or Y or Z

Precision is

282/9997 = 2.82%

407/9997 = 4.07%

512/9997 = 5.12%

ReccomendCount3 counts both the best matching and the second best matching and the third best matching session recommendations.

15010 /1018 = 15 books were recommended on average.

401 /1018 =%39.39 contained X

572 /1018*2 =%28.09 contained X or Y

693 /1018*3 =%22.69 contained X or Y or Z

Precision is

401/15010 = 2.67%

572/15010 = 3.81%

693/15010 = 4.61%

In order to see if the recommendations are any good we will calculate the probability that a random recommender would successfully suggest X,Y and Z.

The probability that a randomly selected book out of the 9544 in the library would be X is 1/9544. The probability that randomly selected 5 books out of 9544 would contain X is given by

$= 1 - \prod_{k=1}^{5}(9544 - k)/9544$

= 1 – [(9543/9544) * (9542/9544) * … *(9538/9544)]

= 1 – [(9543! / 9526!) / 9544^5)]

= 0.001571 = 0.1571% is the expected ratio of successes by a random recommender.

Our experiment yielded 4.32% success rate, so there is a significant improvement although not as much as we would like. However as number of recommended books increase in our system there is a high increase in recall and precision.

In order to visualize the results we will use a 3x3 matrix for our results and another 3x3 matrix for the random book recommender. The tedious calculations for probabilities are not given anymore but the calculation logic is the same as the example given above.

For `cluster set 1` the recall results can be expressed as

Table 9 – Cluster set 1 recall comparison results

C1	Our System				Randomized		
	x	y	z		x	y	z
R1	4.3222	5.500982	6.483301		0.157074	0.209386	0.261676
R2	27.70138	39.98035	50.2947		0.574831	0.679018	0.783106
R3	39.39096	56.18861	68.07466		1.250136	1.405354	1.560345

For `cluster set 1` the precision results can be expressed as

Table 10 – Cluster set 1 precision comparison results

C1	Our System				Randomized		
	x	y	z		x	y	z
R1	0.880528	1.120672	1.320792		0.031999	0.085313	0.159928
R2	2.820846	4.071221	5.121536		0.058535	0.138289	0.239232
R3	2.671552	3.810793	4.616922		0.084786	0.190626	0.317474

6 graphs are shown below. First one is the accuracy rate for r1, r2 and r3. Second one is the same values for randomized recommendation. Third graph is the difference between randomized recommender and our system. Then the graphs are presented for precision values.

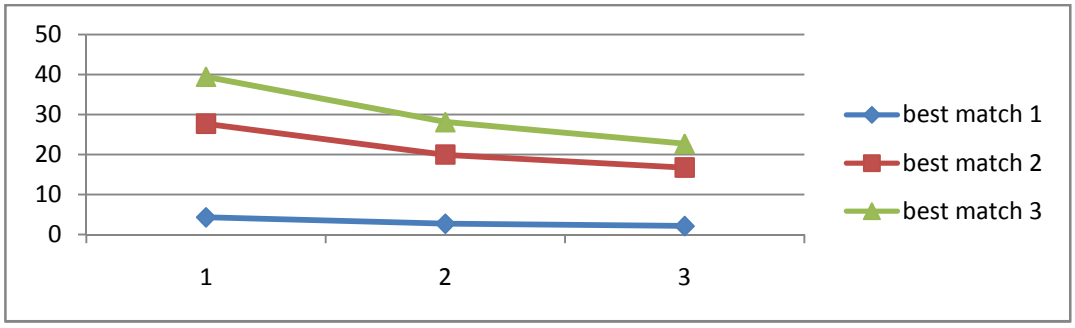

Figure 28 – Cluster set 1 recall values

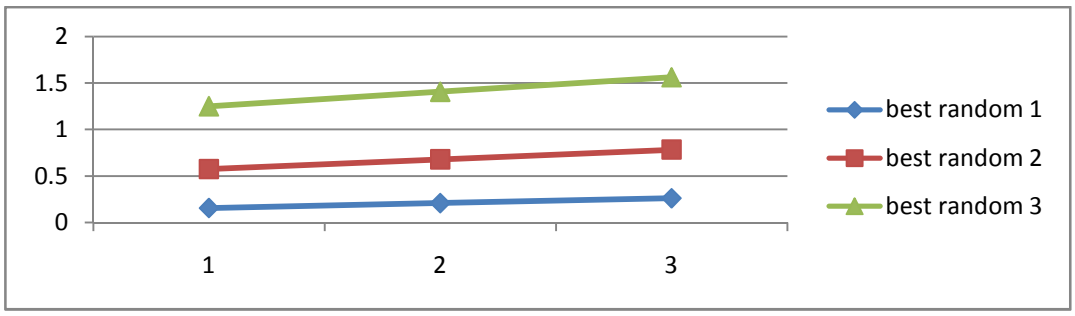

Figure 29 – Random recommender recall values

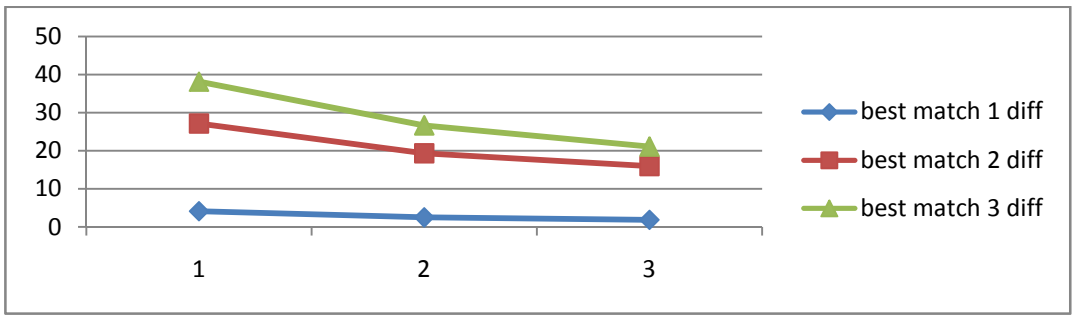

Figure 30 – Cluster set 1 – Results-Random difference

From these results it is seen that the best matching session 3 gives the best results which means recommending more books works in favor of our system. As the number of recommendations increase accuracy is increasing at a faster rate. It is also seen that although more success are obtained if X or Y or Z is counted as a success, it decreases recall but increases accuracy.

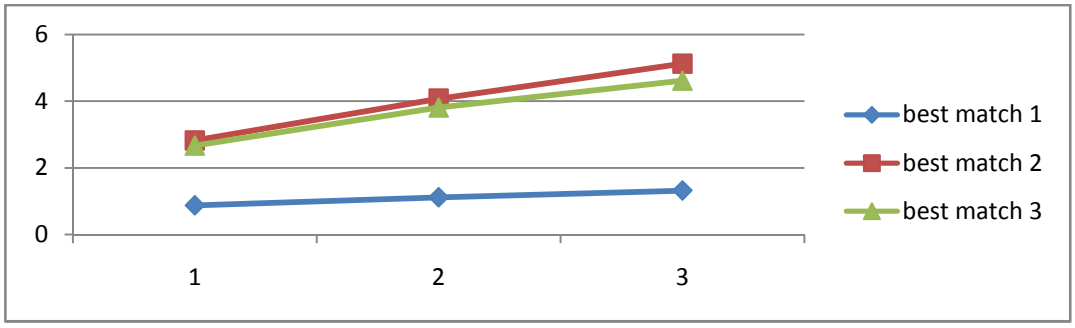

Figure 31 – Cluster set 1 precision values

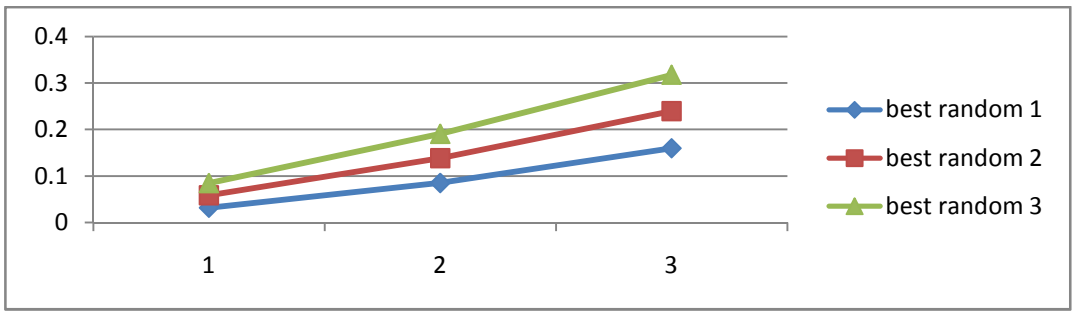

Figure 32 – Random recommender precision values

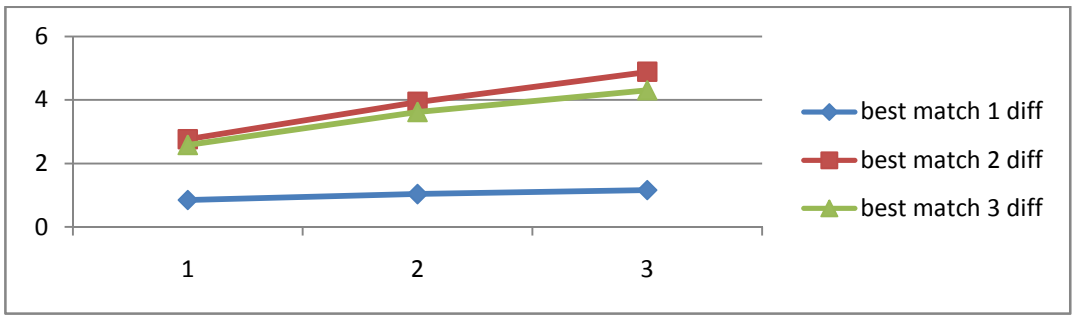

Figure 33 – Cluster set 1 – Results-Random precision difference

It is seen from the resulting graphs that high accuracy differences are obtained compared to a random result generator. It is also seen that just as seen in recall values recommending more books increase precision considerably.

The major increase in both recall and precision are seen when recommendation count per session goes from 5 to 10. Recommendation of 10 books increase recall up to 30% while maintaining a high precision value.

Next experiment is devised to observe the effect of using `recommend 5 most frequent books` part of the system. Next results are obtained by using the same parameters as before however in the recommendation phase instead of recommending the 5 most frequent books in the most similar session, all books in that session are recommended.

Cluster recommendation results are as follows:

time = 03:12:15.8125000

checkedSessionsCount = 1018

reccomendCount1 = 16887 successCount1 = 75 95 114

reccomendCount2 = 33175 successCount2 = 295 421 529

reccomendCount3 = 49517 successCount3 = 410 580 701

As seen recommendation counts have at least tripled. Same computations as before are made and following results are obtained.

Table 11 – Cluster set 2 recall comparison results

C2	Using 5 best				Not Using 5 best		
	x	y	z		x	y	z
R1	4.3222	2.750491	2.1611		7.367387	4.666012	3.732809
R2	27.70138	19.99018	16.7649		28.97839	20.6778	17.32155
R3	39.39096	28.0943	22.69155		40.27505	28.48723	22.9535

Table 12 – Cluster set 2 precision comparison results

C2	Using 5 best				Not Using 5 best		
	x	y	z		x	y	z
R1	0.880528	1.120672	1.320792		0.444129	0.562563	0.675076
R2	2.820846	4.071221	5.121536		0.889224	1.269028	1.594574
R3	2.671552	3.810793	4.616922		0.827998	1.171315	1.415675

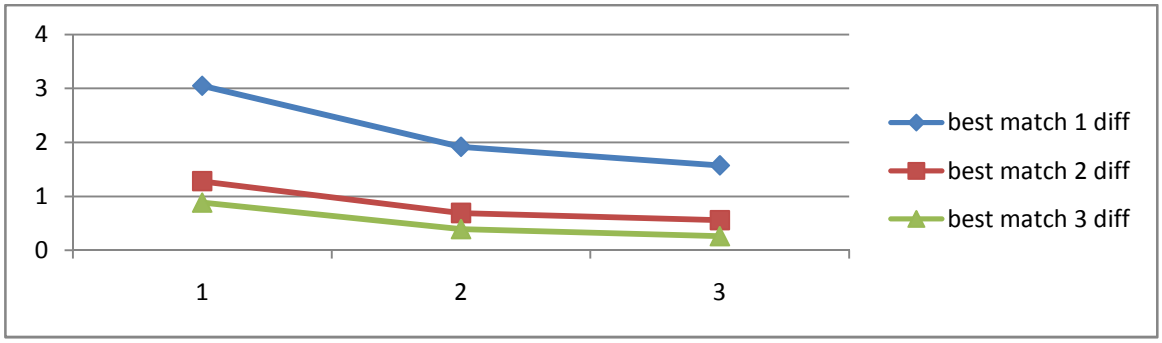

Figure 34 – Cluster set 2 – Use5best-DontUse5Best recall difference

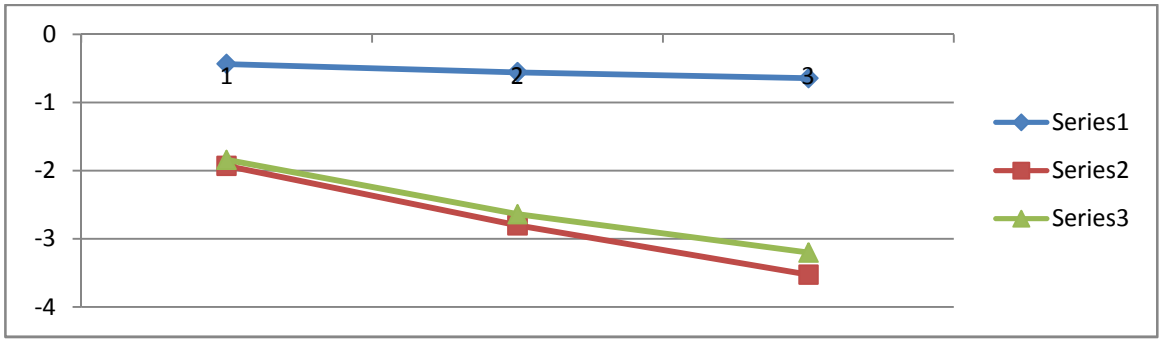

Figure 35 – Cluster set 2 – Use5best-DontUse5Best precision difference

It is easily seen that although little decrease of recall is seen while using only the top 5 frequent books, a huge increase in precision is obtained.

Next experiment is devised to observe the effect of using ontology in our recommendation system. In this control group we will define the distance between

sessions based on whether a book was accessed or not instead of using the ontological distance functions. We will still use the Needleman-Wunsch algorithm however instead of using the distance function we will define similarity of 2 books as

```
If(book1.ID==book2.ID)
        Return 1;
Else
        Return 0;
```

The obtained results are as follows:

time = 00:00:09.6250000
checkedSessionsCount = 1018
reccomendCount1 = 4719 successCount1 = 59 71 84
reccomendCount2 = 9119 successCount2 = 241 318 392
reccomendCount3 = 13466 successCount3 = 315 432 520

Table 13 – Cluster set 3 recall comparison results

C3	Using Ontology				Not Using Ontology		
	x	y	z		x	y	z
R1	4.3222	2.750491	2.1611		5.795678	3.48723	2.750491
R2	27.70138	19.99018	16.7649		23.67387	15.61886	12.83563
R3	39.39096	28.0943	22.69155		30.94303	21.21807	17.02685

Table 14 – Cluster set 3 precision comparison results

C3	Using Ontology				Not Using Ontology		
	x	y	z		x	y	z
R1	0.880528	1.120672	1.320792		1.250265	1.504556	1.780038
R2	2.820846	4.071221	5.121536		2.642834	3.487224	4.298717
R3	2.671552	3.810793	4.616922		2.339225	3.20808	3.861577

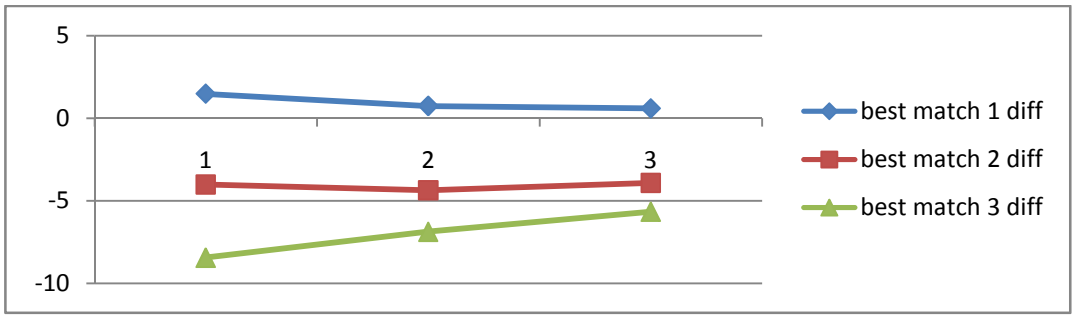

Figure 36 – Cluster set 3 – Use/Don't Use ontology recall difference

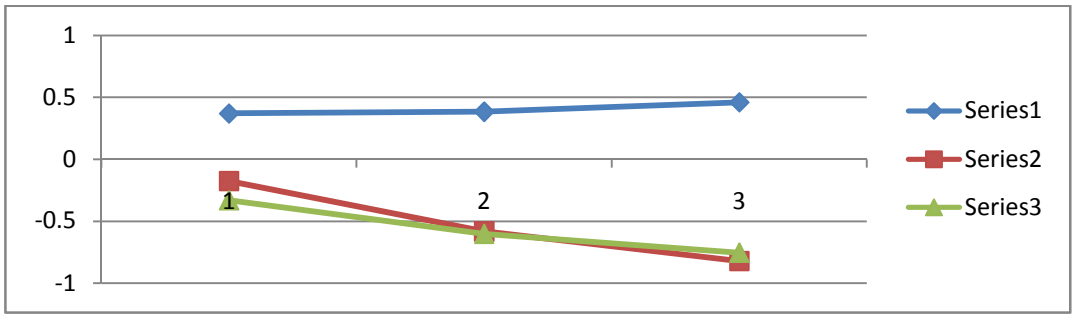

Figure 37 – Cluster set 3 – Use/Don't Use ontology precision difference

Although in the case where only 5 books are recommended not using ontology has an advantage, when recommendation count is increased to 10 this advantage is reversed. High recall/precision advantages are seen with using an ontology for distance calculations.

Next experiment is devised to observe the effect of using temporal sequence in our recommendation system. In this control group we will define the distance between sessions as the distance of `set of ontological objects` instead of using the `sequence of ontological objects`. We will not use the Needleman-Wunsch algorithm. The distance of the whole set is calculated as the average distance of all items in two sets.

The obtained results are as follows:

time = 00:25:01.6875000

checkedSessionsCount = 1018

reccomendCount1 = 2123 successCount1 = 23 29 34

reccomendCount2 = 4361 successCount2 = 91 124 156

reccomendCount3 = 6691 successCount3 = 134 184 231

Table 15 – Cluster set 4 recall comparison results

C4	Using Sequence				Not Using Sequence		
	x	y	z		x	y	z
R1	4.3222	2.750491	2.1611		2.259332	1.424361	1.113294
R2	27.70138	19.99018	16.7649		8.939096	6.090373	5.108055
R3	39.39096	28.0943	22.69155		13.16306	9.037328	7.563851

Table 16 – Cluster set 4 precision comparison results

C4	Using Sequence				Not Using Sequence		
	x	y	z		x	y	z
R1	0.880528	1.120672	1.320792		1.083373	1.365992	1.601507
R2	2.820846	4.071221	5.121536		2.086677	2.843385	3.577161
R3	2.671552	3.810793	4.616922		2.00269	2.749963	3.452399

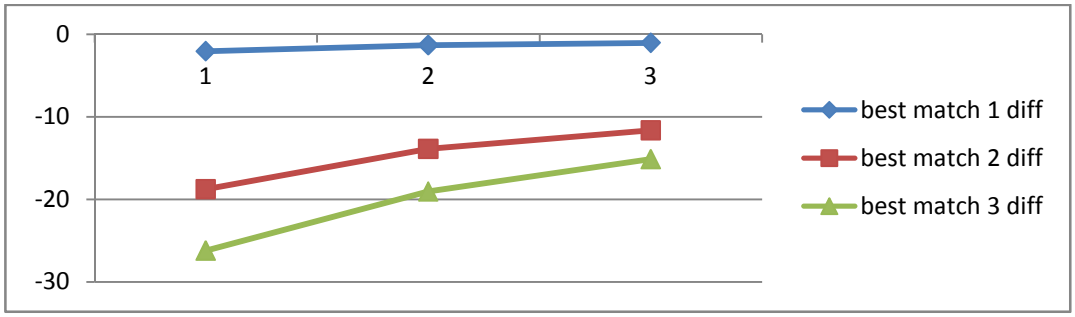

Figure 38 – Cluster set 4 – Use/Don't Use Needleman-Wunsch recall difference

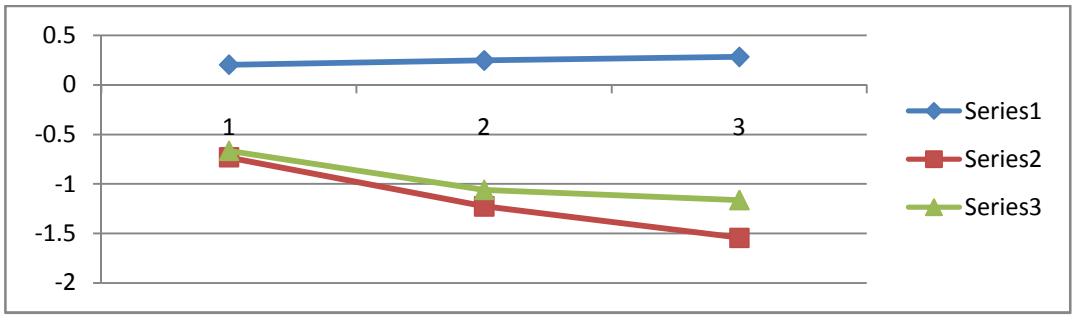

Figure 39 – Cluster set 4 – Use/Don't Use Needleman-Wunsch precision difference

Again although in the case where only 5 books are recommended not using the sequence has an advantage, when recommendation count is increased to 10 this advantage is reversed. High recall/precision advantages are seen with using the Needleman-Wunsch algorithm for distance calculations.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work we have proposed a working system for online recommendation using ontologies and ontology object clustering. We have implemented all phases of the system which are mainly data acquisition, ontology creation, data mining/clustering and recommendation. We have devised and run several experiments to see the impact of the chosen parameters of the system on the results and the accuracy values they generate.

The major points of our work have been integrating semantics into web usage mining, using clustering on a semantic level in order to capture user access patterns to a website and making use of the temporal structure of these accesses. Also the design of the system allows the system to be unaffected by the `new item problem` that effects most rating based systems.

For evaluation purposes we have calculated how accurate a randomized recommender would perform under the same conditions and compared our system to it. Also we have devised experiments to see how well the same system would work if semantics or temporal structure were discarded and simpler methods were used.

We have seen that our system reaches 40% recall values where precision is around 3%. It is noted that a random recommender would do magnitudes less in the same conditions, around 1.25% recall at 0.08% precision. We have also noted in our experiments that, at around 10 book recommendations per session, using

semantics and temporal coherence of sequences greatly increases recall and precision values.

6.2 Future Work

We have seen in our work that using semantic information for user access pattern generation greatly improves recall and precision. However the dataset we have used was restrictive so only a small ontology could be used in our experiments. A larger dataset with more possibilities of ontological objects can reveal greater benefits of using semantic knowledge.

One other restriction of the dataset was that it contains only around ten thousand books in it so it can be said to be a sparse dataset. Clustering based systems usually give better results using dense datasets. A dense dataset can be used to improve clustering results.

We have also seen that defining user sessions as a sequence of ontological objects instead of a simple set improves accuracy. We have used the Needleman-Wunsch algorithm for implementation with gap penalty of 0. Using different gap penalties could have an effect on the results. Effects of varying gap penalties could be observed and possibly improvements could be observed by doing so.

One unresolved issue in this study has been the method of finding the mean of the clusters in clustering phase. Although the current method works, a multi-sequence matching algorithm can be used to utilize the usage of the sequence matching distance metric already used in the system. Multi-sequence matching algorithms are very slow in nature and optimal solutions are hard to find. A different method could be devised to find a middle ground between the current method and such algorithms.

REFERENCES

[1] Chaim Zins, Conceptual approaches for defining data, information, and knowledge: Research Articles Journal of the American Society for Information Science and Technology Volume 58 , Issue 4 (February 2007)

[2] John Wang, Data Mining: Opportunities and Challenges
IGI Global; illustrated edition (February 4, 2003)

[3] Cooley, R. and Mobasher, B. and Srivastava, J. (1997) Web mining: Information and pattern discovery on the world wide web. In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), Los Alamitos .

[4] da Costa, M.G., Jr. Zhiguo Gong, Web Structure Mining: An Introduction nformation Acquisition, (2005) IEEE International Conference on. DOI: 10.1109/ICIA.2005.1635156

[5] Sanjay Kumar Madria Sourav S. Bhowmick Wee Keong Ng Ee-Peng Lim (1999) Research Issues in Web Data Mining Lecture Notes In Computer Science; Vol. 1676 Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery Pages: 303 - 312

[6] Ellen Spertus (1997) ParaSite: mining structural information on the Web Computer Networks and ISDN Systems archive Volume 29 , Issue 8-13 September 1997) table of contents Pages: 1205 – 1215

[7] L. Lakshmanan, F. Sadri, and I. N. Subramanian. A declarative language for querying and restructuring the web. In Proc. 6th International Workshop on Research Issues in Data Engineering: Interoperability of Nontraditional Database Systems (RIDE-NDS'96), 1996.

[8] Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry (1999) The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. Stanford InfoLab

[9] Jon M. Kleinberg Cornell Univ., Ithaca, NY Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) Volume 46 , Issue 5 (September 1999) Pages: 604 - 632

[10] T. Srivastava, P. Desikan, V. Kumar Web Mining – Concepts, Applications and Research Directions Foundations and Advances in Data Mining (2005), pp. 275-307

[11] Jaideep Srivastava Robert Cooley Mukund Deshpande Pang-Ning Tan Web usage mining: discovery and applications of usage patterns from Web data ACM SIGKDD Explorations Newsletter Volume 1, Issue 2 (January 2000) COLUMN: Survey articles Pages: 12 - 23

[12] Log Files - Apache HTTP Server http://eregie.premier-ministre.gouv.fr/manual/logs.html Last Access: 1/21/2010

[13] HTTP/1.1: Status Code Definitions http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html Last Access: 1/21/2010

[14] Pang-Ning Tan, Vipin Kumar (2001). Discovery of Web robot sessions based on their navigational patterns. Data Mining and Knowledge Discovery Volume 6, Issue 1 (January 2002) Pages: 9 - 35

[15] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava (1999). Data preparation for mining World Wide Web browsing patterns *Knowledge and Information Systems*, Vol. 1, No. 1. (1999), pp. 5-32

[16] A statistical mining tool. WebLogExpert http://www.weblogexpert.com/ Last Accessed: 1/21/2010

[17] A statistical mining tool. Nihuo Software http://www.loganalyzer.net/ Last Access: 1/21/2010

[18] A statistical mining tool. http://www.log-analyzer.net/ Last Access: 1/21/2010

[19] A statistical mining tool. http://www.google.com/analytics/ Last Access: 1/21/2010

[20] Alex Berson, Stephen Smith, and Kurt Thearling (2002) Building Data Mining Applications for CRM ISBN: 0071372717 McGraw-Hill, Inc. New York, NY, USA

[21] Piatetsky-Shapiro, G. (1991), Discovery, analysis, and presentation of strong rules, in G. Piatetsky-Shapiro & W. J. Frawley, eds, 'Knowledge Discovery in Databases', AAAI/MIT Press, Cambridge, MA.

[22] R. Agrawal; T. Imielinski; A. Swami: Mining Association Rules Between Sets of Items in Large Databases", SIGMOD Conference 1993: 207-216

[23] Agrawal R., Srikant R., "Fast Algorithms for Mining Generalized Association Rules", Proceedings of the 20th International Conference on Very Large Databases (VLDB'94), Santiago, Chile, September 1994.

[24] Brin S., Motwani R., Ullman J., Tsur S., "Dynamic Itemset Counting and Implication Rules for Market Basket Data", Proceedings of the International Conference on Management of Data (SIGMOD'97), Tucson, Arizona, May 1997, p. 255-264.

[25] Fayad U., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., Eds., Advances in Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 1996.

[26] Savasere A., Omiecinski E., Navathe S., "An Efficient Algorithm for Mining Association Rules in Large Databases", Proceedings of the 21 st International

Conference on Very Large Databases (VLDB'95), Zurich, Switzerland, September 1995, p. 432-444.

[27] Toivonen H., "Sampling Large Databases for Association Rules", Proceedings of the 22nd International Conference on Very Large Databases (VLDB'96), September 1996.

[28] Agrawal R., Srikant R., "Mining Sequential Patterns", Proceedings of the 11th International Conference on Data Engineering (ICDE'95), Tapei, Taiwan, March 1995.

[29] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Proceedings of the Fifth International Conference on Extending Database Technology, Avignon, France, 1996.

[30] Masseglia F., Cathala F., Poncelet P., "The PSP Approach for Mining Sequential Patterns", Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98), LNAI, Vol. 1510, Nantes, France, September 1998, p. 176-184

[31] Mohammed J. Zaki SPADE: An Efficient Algorithm for Mining Frequent Sequences Machine Learning, Vol. 42, No. 1/2. (2001), pp. 31-60.

[32] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu. Freespan: Frequent pattern-projected sequential pattern mining. In Proc. 2000 Int. Conf. Knowledge Discovery and Data Mining (KDD'00), pages 355–359,Boston, MA, Aug. 2000.

[33] , J. Han, Mortazavi B. Asl, H. Pinto, Q. Chen, U. Dayal, M. C. Hsu PrefixSpan Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth Export by: J. PeiIn Proc.17th Int'l Conf. on Data Eng. (2001), pp. 215-226.

[34] Xifeng Yan, Jiawei Han, Ramin Afshar CloSpan: Mining Closed Sequential Patterns in Large Datasets In In SDM (2003), pp. 166-177.

[35] A. K. Jain, M. N. Murty, P. J. Flynn Data clustering: a review ACM Comput. Surv. Vol. 31, No. 3. (September 1999), pp. 264-323.

[36] Jain, A. K. And Dubes, R. C. 1988. Algorithms for Clustering Data. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ. ISBN-10: 013022278X

[37] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, Stuart Russell Distance metric learning, with application to clustering with side-information In Advances in Neural Information Processing Systems 15, Vol. 15 (2002), pp. 505-512

[38] Tsang, I. W., & Kwok, J. T. (2003). Distance metric learning with kernels. International Conference on Artificial Neural Networks (pp. 126-129)

[39] Pavel Berkhin Survey of Clustering Data Mining Techniques / Grouping Multidimensional Data (2002) Technical Report Accrue Software

[40] George Karypis, Eui-Hong (Sam) Han, Vipin Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling," *Computer*, vol. 32, no. 8, pp. 68-75, Aug. 1999, doi:10.1109/2.781637

[41] Daniel Boley Principal Direction Divisive Partitioning Data Mining and Knowledge Discovery archive Volume 2 , Issue 4 (December 1998) Pages: 325 - 344

[42] Andrew W. Moore Very fast EM-based mixture model clustering using multiresolution kd-trees Proceedings of the 1998 conference on Advances in neural information processing systems II table of contents Pages: 543 - 549

[43] Hartigan, J. 1975. Clustering Algorithms. John Wiley & Sons, New York, NY.

[44] Hartigan, J. And Wong, M. 1979. Algorithm AS136: A k-means clustering algorithm. Applied Statistics, 28, 100-108

[45] Kaufman, L. And Rousseeuw, P. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990.

[46] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics 20: 53–65.

[47] Jörg Sander Martin Ester Hans-Peter Kriegel Xiaowei Xu Density-Based Clustering in Spatial Databases:The Algorithm GDBSCAN and its Applications Data Mining and Knowledge Discovery Volume 2 , Issue 2 (June 1998) Pages: 169 - 194

[48] Information Management: A Proposal Tim Berners-Lee, CERN March 1989, May 1990

[49] RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, R. Fielding, J. Getty, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee (June 1999) http://www.ietf.org/rfc/rfc2616.txt Last Access: 1/21/2010

[50] Tim Berners-Lee, James Hendler and Ora Lassila The Semantic Web Scientific American: Feature Article: The Semantic Web: May 2001

[51] Definition of Semantics http://www.merriam-webster.com/dictionary/Semantics Last Access: 1/21/2010

[52] Rudolf Carnap An Introduction to the Philosophy of Science Dover Publications (January 17, 1995) ISBN-10: 0486283186

[53] Karl Popper The Logic of Scientific Discovery Routledge; New edition edition (March 29, 2002) ISBN-10: 0415278449

[54] The Semantic Web in Breadth by Aaron Swartz http://logicerror.com/semanticWeb-long Last Access: 1/21/2010

[55] The Semantic Web: An Introduction by Sean B. Palmer, 2001-09 http://infomesh.net/2001/swintro/#simpleData Last Access: 1/21/2010

[56] "Semantic Web - XML2000, slide 10". W3C Semantic Web Stack http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html Last Access: 1/21/2010

[57] RFC 3986 / STD 66 (2005) – the current generic URI syntax specification http://www.ietf.org/rfc/rfc3986.txt Last Access: 1/21/2010

[58] List of URI schemas - http://www.w3.org/Addressing/schemes Last Access: 1/21/2010

[59] XML specifications http://www.w3.org/XML/ Last Access: 1/21/2010

[60] Resource Description Framework (RDF) Model and Syntax Specification http://www.w3.org/TR/PR-rdf-syntax/ Last Access: 1/21/2010

[61] Resource Description Framework (RDF): Concepts and Abstract Syntax http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ Last Access: 1/21/2010

[62] Resource Description Framework (RDF) Schema Specification 1.0 W3C Candidate Recommendation 27 March 2000 http://www.w3.org/TR/2000/CR-rdf-schema-20000327/ Last Access: 1/21/2010

[63] Notation3 specification by w3c http://www.w3.org/DesignIssues/Notation3 Last Access: 1/21/2010

[64] RDF Vocabulary Description Language 1.0: RDF Schema http://www.w3.org/TR/rdf-schema/#ch_resource Last Access: 1/21/2010

[65] Thomas R. Gruber A Translation Approach to Portable Ontology Specifications Knowledge Acquisition Volume 5 , Issue 2 (June 1993) Special issue: Current issues in knowledge modeling Pages: 199 - 220

[66] Pizza ontology http://www.co-ode.org/ontologies/pizza/2007/02/12/pizza.owl Last Access: 1/21/2010

[67] The Protégé Ontology Editor and Knowledge Acquisition System

http://protege.stanford.edu/ Last Access: 1/21/2010

[68] OWLPropViz – A property relation visualization pluging for Protege http://protegewiki.stanford.edu/index.php/OWLPropViz Last Access: 1/21/2010

[69] Jérôme Euzenat, Pavel Shvaiko Ontology Matching (2007) Springer-Verlag New York, Inc. Secaucus, NJ, USA ISBN: 3540496114

[70] Eidoon, Z. Yazdani, N. Oroumchian, F. A Vector Based Method of Ontology Matching In Proceedings of SKG, 2007.

[71] 4. P. Patel-Schneider, P. Hayes, I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax http://www.w3.org/TR/2003/WD-owl-semantics-20030331/ Last Access: 1/21/2010

[72] Dieter Fensel, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, "OIL: An Ontology Infrastructure for the Semantic Web," IEEE Intelligent Systems, vol. 16, no. 2, pp. 38-45, Mar./Apr. 2001, doi:10.1109/5254.920598

[73] OWL Web Ontology Language Overview http://www.w3.org/TR/owl-features/ Last Access: 1/21/2010

[74] OWL 2 Web Ontology Language Primer http://www.w3.org/TR/owl2-primer/ Last Access: 1/21/2010

[75] Hypertext & Information Retrieval & Web Mining http://www.cyberartsweb.org/cpace/ht/lanman/up1.htm by Lan Man. Last Access: 1/21/2010

[76] An Extensive Survey of Clustering Methods for Data Mining www.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf Last Access: 1/21/2010

[77] Shahabi, C., Zarkesh, A. M., Adibi, J., and Shah, V., Knowledge discovery from users Web-page navigation. In Proceedings of Workshop on Research Issues in Data Engineering, Birmingham, England, 1997.

[78] Nasraoui, O., Frigui, H., Joshi, A., Krishnapuram, R., Mining Web access logs using relational competitive fuzzy clustering. To appear in the Proceedings of the Eight International Fuzzy Systems Association World Congress, August 1999.

[79] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," Pattern Recognition, vol. 30, No. 7, pp. 1109-1119, 1997.

[80] Bamshad Mobasher, Robert Cooley, Jaideep Srivastava Automatic personalization based on Web usage mining Communications of the ACM, Vol. 43, No. 8. (2000), pp. 142-151.

[81] Han, E-H, Karypis, G., Kumar, V., and Mobasher, B., Hypergraph based clustering in high-dimensional data sets: a summary of results. IEEE Bulletin of the Technical Committee on Data Engineering, (21) 1, March 1998.

[82] B. Mobasher, R. Cooley, J. Srivastava Creating adaptive Web sites through usage-based clustering of URLs Knowledge and Data Engineering Exchange, 1999. (KDEX '99) Proceedings. 1999 Workshop on In Knowledge and Data Engineering Exchange, 1999. (KDEX '99) Proceedings. 1999 Workshop on (1999), pp. 19-25.

[83] Yanchun Zhang, Guandong Xu, Xiaofang Zhou: A Latent Usage Approach for Clustering Web Transaction and Building User Profile. ADMA 2005: 31-42

[84] Bamshad Mobasher , Hoghua Dai , Tao Luo , Yuqing Sun , Jiang Zhu Integrating Web Usage and Content Mining for More Effective Personalization (2000) In E-Commerce And Web Technologies Lecture Notes In Computer Science

[85] Bamshad Mobasher , Hoghua Dai , Tao Luo , Yuqing Sun , Jiang Zhu Integrating Web Content Mining into Web Usage Mining for Finding Patterns and Predicting Users' Behaviors International Journal of Information Science and Management, Vol, 7, No. 1 January / June, 2009

[86] Kohonen, T. (1995). Self-Organizing Maps. Series in Information Sciences, Vol. 30. Springer, Heidelberg. Second ed. 1997.

[87] Paola Britos, Damián Martinelli, Hernán Merlino, Ramón García-Martínez Web Usage Mining Using Self Organized Maps IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.6, June 2007

[88] Abraham, A. Ramos, V. (2003)Web usage mining using artificial ant colony clustering and linear genetic programming Genetic Programming, Congress on Evolutionary Computation (CEC), IEEE Pages 1384--1391

[89] Ramos V., Muge F., Pina P., Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies, Soft Computing Systems - Design,Management and Applications, 2nd Int. Conf. on Hybrid Intelligent Systems, IOS Press, pp. 500-509, 2002.

[90] Dai, H., & Mobasher, B. (2002). Using ontologies to discover domain-level web usage profiles. Proceedings of the 2nd Semantic Web Mining Workshop at ECML/PKDD 2002. Helsinki, Finland.

[91] Bettina Berendt, Andreas Hotho, Gerd Stumme Towards Semantic Web Mining Lecture Notes In Computer Science; Proceedings of the First International Semantic Web Conference on The Semantic Web table of contents Pages: 264 - 278 Year of Publication: 2002

[92] Rokia Missaoui, Petko Valtchev, Chabane Djeraba, Mehdi Adda Toward Recommendation Based on Ontology-Powered Web-Usage Mining IEEE Internet Computing Volume 11 , Issue 4 (July 2007) Pages: 45-52

[93] http://protege.cim3.net/file/pub/ontologies/travel/travel.owl Travel ontology Last Access: 2/2/2010

[94] Suleyman Salin, Pinar Senkul: (2009) Using semantic information for web usage mining based recommendation. ISCIS 2009: 236-241

[95] Jike Ge Yuhui Qiu Concept Similarity Matching Based on Semantic Distance Proceedings of the 2008 Fourth International Conference on Semantics, Knowledge and Grid Pages: 380-383

[96] Alexander Maedche Valentin Zacharias Clustering Ontology-Based Metadata in the Semantic Web Lecture Notes In Computer Science; Vol. 2431 Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery Pages: 348 - 360

[97] Needleman SB, Wunsch CD. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". J Mol Biol 48 (3): 443–53. doi:10.1016/0022-2836(70)90057-4. PMID 5420325

APPENDIX A – Sample Book RDF

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/0.1/foaf/"
      xmlns:SeckinConcepts="http://www.seckin.com.tr/concepts.owl#
>
      <rdf:Description
rdf:about="http://www.seckin.com.tr/yazarlar.owl#Norman_Vincent_Pe
ale">
      <SeckinConcepts:isYazarOf>
            <SeckinConcepts:Kitap
rdf:ID="Olumlu_Yaşamanın_Gücü2679">
                  <SeckinConcepts:hasKategori>
                        <rdf:Description
rdf:about="http://www.seckin.com.tr/Kategori.owl#Psikoloji_191200_
1">
                        <SeckinConcepts:isKategoryOf
rdf:resource="#Olumlu_Yaşamanın_Gücü2679"/>
                        </rdf:Description>
                  </SeckinConcepts:hasKategori>
                  <SeckinConcepts:hasYazar
rdf:resource="http://www.seckin.com.tr/yazarlar.owl#Norman_Vincent
_Peale"/>
                  <SeckinConcepts:Aciklama
rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >/ İnançlı
Olmak Başarılı Olmaktır / Kendinizden Çıkın / İçinizdeki Boşluğu
Doldurun / Korkuyu Yenelim / Eski Gücünüze Kavuşma / Başarının
Ebedi Sırrı</SeckinConcepts:Aciklama>
                  <SeckinConcepts:SayfaSayisi
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">252</SeckinCon
cepts:SayfaSayisi>
```

```
                <SeckinConcepts:AltAciklama
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"></SeckinCon
cepts:AltAciklama>
                <SeckinConcepts:ISBN
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">975 322 020
0</SeckinConcepts:ISBN>
                <SeckinConcepts:KargoyaVerisGun
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</SeckinConce
pts:KargoyaVerisGun>
                <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2679</rdfs:
comment>
                <SeckinConcepts:UstAciklama
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"></SeckinCon
cepts:UstAciklama>
                <SeckinConcepts:hasKitapFiyat>
                      <SeckinConcepts:KitapFiyat
rdf:ID="KitapFiyat_2679">
                            <SeckinConcepts:ParaBirimi
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>YTL</SeckinConcepts:ParaBirimi>
                            <SeckinConcepts:KDV
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0</SeckinConcepts:KDV>
                            <SeckinConcepts:Indirim
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0.00</SeckinConcepts:Indirim>
                            <SeckinConcepts:isKitapFiyatOf
rdf:resource="#Olumlu_Yaşamanın_Gücü2679"/><SeckinConcepts:Fiyat1
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>0.00</SeckinConcepts:Fiyat1>
                            <SeckinConcepts:Fiyat2
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>9500000.00</SeckinConcepts:Fiyat2>
                      </SeckinConcepts:KitapFiyat>
                </SeckinConcepts:hasKitapFiyat>
```

<SeckinConcepts:UrunSira rdf:datatype="http://www.w3.org/2001/XMLSchema#int" >5</SeckinConcepts:UrunSira>
<SeckinConcepts:Spot rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >İnsan sonsuz bir güç odağıdır. Kişinin kendi güçsüzlüğüne ve yapamayacaklarına inanması bu güce set koyar. Olumluya inanmak bu gücü kullanabileceğimiz alanı genişletmektedir. Sağduyu, yani içimizdeki ses bu sonsuzluk içinde bizim için en doğru yönü fısıld</SeckinConcepts:Spot>
<SeckinConcepts:Ebat rdf:datatype="http://www.w3.org/2001/XMLSchema#string">13.5×19.5 cm.</SeckinConcepts:Ebat>
<SeckinConcepts:YayinYili rdf:datatype="http://www.w3.org/2001/XMLSchema#string" >2003</SeckinConcepts:YayinYili>
<SeckinConcepts:hasYayinEvi rdf:resource="http://www.seckin.com.tr/YayinEvi.owl#Tüdav_Yayınları"/>
<SeckinConcepts:hasDefaultKategory rdf:resource="http://www.seckin.com.tr/Kategori.owl#Psikoloji_191200_1"/>
</SeckinConcepts:Kitap>
</SeckinConcepts:isYazarOf>
</rdf:Description>
</rdf:RDF>