# IMPROVING THE PREDICTION OF PAGE ACCESS BY USING SEMANTICALLY ENHANCED CLUSTERING

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

BY

ERMAN ŞEN

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

SEPTEMBER 2014

Approval of the thesis:

**IMPROVING THE PREDICTION OF PAGE ACCESS BY USING SEMANTICALLY ENHANCED CLUSTERING**

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

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

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

Prof. Dr. İsmail Hakkı Toroslu ____________________
Supervisor, **Computer Engineering, METU**

Assoc. Prof. Dr. Pınar Karagöz ____________________
Co-supervisor, **Computer Engineering, METU**

**Examining Committee Members:**

Prof. Dr. Ahmet Coşar ____________________
Computer Engineering Department, METU

Prof. Dr. İsmail Hakkı Toroslu ____________________
Computer Engineering Department, METU

Assoc. Prof. Dr. Pınar Karagöz ____________________
Computer Engineering Department, METU

Assoc. Prof. Dr. Halit Oğuztüzün ____________________
Computer Engineering Department, METU

Assist. Prof. Dr. Alev Mutlu ____________________
Computer Engineering Department, Kocaeli University

Date: ____________________

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

|  |  |  |
|---|---|---|
| Name, Last Name | : | ERMAN ŞEN |
| Signature | : |  |

# ABSTRACT

IMPROVING THE PREDICTION OF PAGE ACCESS BY USING SEMANTICALLY ENHANCED CLUSTERING

Şen, Erman

M.Sc., Department of Computer Engineering

Supervisor : Prof. Dr. İsmail Hakkı Toroslu

Co-Supervisor : Assoc. Prof. Dr. Pınar Karagöz

September 2014, 73 pages

There are many parameters that may affect the navigation behaviour of web users. Prediction of the potential next page that may be visited by the web user is important, since this information can be used for prefetching or personalization of the page for that user. One of the successful methods for the determination of the next web page is to construct behaviour models of the users by clustering. The success of clustering is highly correlated with similarity measure that is used for calculating the similarity among navigation sequences. This thesis proposes a new approach for determining the next web page by extending the standard clustering method with the content-based semantic similarity method. The success of the proposed method has also been shown through real life web log data.

Keywords – Ontology, concept set similarity, session similarity, sequence alignment.

# ÖZ

ANLAMSAL GELİŞMİŞ SINIFLANDIRMA İLE GELİŞMİŞ SAYFA ERİŞİM TAHMİNİ GELİŞTİRME

Şen, Erman

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. İsmail Hakkı Toroslu

Ortak Tez Yöneticisi : Doç. Dr. Pınar Karagöz

Eylül 2014, 73 sayfa

Web kullanıcılarının navigasyon davranışını etkileyebilir birçok parametre mevcuttur. İnternet kullanıcıları tarafından ziyaret edilebilir potansiyel bir sonraki sayfanın tahmin edilebilirliği önemlidir, çünkü bu bilgiler, söz konusu kullanıcı için ön yükleme veya sayfanın kişiselleştirilebilmesi için kullanılabilir. Bir sonraki web sayfasının belirlenmesi için başarılı yöntemlerden biri de kümeleme ile kullanıcıların davranış modelleri inşa etmektir. Kümelenme başarısı, navigasyon dizileri arasında benzerlik hesaplamak için kullanılan benzerlik ölçüsü ile doğrudan ilişkilidir. Bu tez, içerik tabanlı anlamsal benzerlik yöntemi ile standart kümeleme yöntemini genişletmek suretiyle bir sonraki web sayfasını belirlemek için yeni bir yaklaşım önermektedir. Önerilen yöntemin başarısı da gerçek hayat web günlüğü verileriyle kanıtlanmıştır.

Anahtar kelimeler – varlıkbilim, kavram küme benzerliği, oturum benzerliği, dizi hizalama.

*to Eren, Erdil and Dilek*

# ACKNOWLEDGMENTS

I would like to express my sincere appreciation and thanks to my supervisor Prof. Dr. İsmail Hakkı Toroslu and my co-supervisor Assoc. Prof. Dr. Pınar Karagöz for encouraging my research, and for leading me to complete my study. Thank you for your guidance, continuous support, patience, valuable time and insight throughout the study. I would also like to thank to my committee members, Prof. Dr. Ahmet Coşar, Assoc. Prof. Dr. Halit Oğuztüzün and Assist. Prof. Dr. Alev Mutlu for serving as my committee members as well as for their suggestions and comments. I also want to thank you all for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions.

A special thanks to my family. Words cannot express how grateful I am to my sons Eren and Erdil for their understanding my frequent absences and for all of the sacrifices that they have made on my behalf. At the end, I would like to express special appreciation to my beloved wife Dilek Şen. I would like to thank for her encouragement and willingness to endure me, for her support in the moments when I felt depressed and for spending sleepless nights with.

# TABLE OF CONTENTS

CHAPTERS

APPENDICES

# LIST OF TABLES

## TABLES

# LIST OF FIGURES

## FIGURES

# CHAPTER 1

# INTRODUCTION

Recently, the growth of the World Wide Web has become tremendous being the very first reference as an information resource. Although the data available in web is so divergent and unstructured, effective new tools and methods have been developed to enable end-users access their targets easily. Through – so called "data mining" techniques, it is now just a matter of a click to find the relevant content out of that enormous available data.

In spite of this fact, however, due to the information overload in web, it becomes more important to be able to recognize the web users' behaviours on the web sites, and improve the performances of web sites and the qualities of user experiences on these sites by using techniques like web page personalization and increasing the web page download time using techniques like prefetching. Recommendation and prediction systems emerge at this point for increasing the relevancy of the content searched for. These systems allow users in personalizing and customizing their own environment [1].

The data that is processed to predict the next page of web user is her previous visit logs. However, as in many log processing problems, this problem also suffers from the data sparsity. Usually there are not so much log records for users, and each session of user on a web site might have different purpose. Therefore, it may become impossible to determine the potential next page of web user by only looking at her previous records, but rather, the navigation behaviours of similar users, with similar purposes, should also be considered. Furthermore, rather than just processing the URL's of the web pages, the purpose of the visit of the web user may be determined better by also processing of the content of the page.

A useful solution against to this sparsity problem is to group the items in terms of concepts and to identify common items by using their semantic similarity instead of exact matching. In this work, we present a new recommendation approach by constructing a *cluster-based* model, which is based on semantic similarity of web page content. In the pre-processing phase, each web page is re-represented as a set of concepts extracted from the content. In accordance with it, page access sessions of users are reconstructed as a sequence of concept sets. The hearth of the proposed method lies in finding the similarity between two sessions, which are actually sequence of concept sets. The length of the sequences may vary as the number of pages navigated by different users generally vary. After the model is constructed, the recommendation is achieved via fetching the most similar cluster to the partial session under interest and finding *k-nn* items in the cluster.

The experimental evaluation is conducted on web logs of a real web server. The results reveal the advantage of the proposed method over similar approaches that do not involve semantic similarity.

This thesis is organized as follows; Chapter 2 gives background for web mining and related concepts. Chapter 3 introduces the method, model construction and the prediction process together with some preliminary tasks and basic definitions. In Chapter 4, the proposed method is explored by testing the "*concept set similarity*" on the dataset and alternative options are evaluated within the scope of the approach. In the last chapter, contributions are explored and some ideas for future work are stated.

# CHAPTER 2

# BACKGROUND

Predicting users' web page navigation sequences is a challenging task in nowadays technology world and is important towards improving the accuracy rate for the information searched. Predicting users' next page access helps much for web site personalization [2]. Erinaki et al. define the web personalization as the customization of a web site for specific needs for each user through the analysis of the their navigational behavior [3]. They also underlined the fact that integration of usage, structure or user profile enhances the personalization results.

The main objective of Web mining and Web personalization is to extract meaningful navigation sequences for foreseeing user's navigational behaviours and then make use of it for designing better recommendation and prediction systems. In addition, understanding users' navigational behaviours also provides input to web sites to be revised dynamically in accordance with user needs [4,5,6]. An in depth study for Web mining and Web personalization can be found in [7].

Mobasher et al. [8,9] proposes a framework – WebPersonalizer – in extracting crucial information to be used in prediction systems by utilizing users' navigation behaviours. They also extended their study by including user profiles into their framework [10,11]. They merged content and user profiles and used them to associate web page views in a balanced conjunction manner.

Berendt et al. [12] defined “concept hierarchy” to be used in analyzing user’ navigation patterns. They concentrated on navigation rather than the content retrieved. Their study was focused on representing Web pages as high level of concept instances by which they were able to analyze the navigation patterns instead of proposing a recommendation system.

The upcoming sections of this chapter highlights the basic components of Web mining concept together with their sub-categories.

## 2.1 Web mining

From a broad perspective, the definition of Web mining can be given as gathering useful information by using resources available in web. Keeping in mind that resources are so dynamic, unstructured and big in volume, special techniques particular to data mining have been developed.

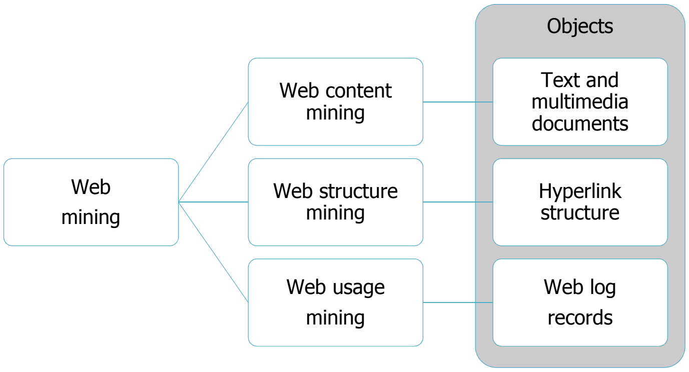


*Figure 2.1. Web mining classification and objects*

Furthermore, Web mining can be studied under different topics depending on the purpose or the intention of the research under consideration. In [13], Chen classified it under three distinct grouping; content mining, structure mining and usage mining. Figure 2.1 depicts this classification together with their associated objects.

### 2.1.1 Web content mining

Capturing and extracting valuable information and/or knowledge from the content of web pages have been an important research area within the scope of data mining. Since the web is a huge collection of resources, two common characteristics of the web have to be taken into consideration; first the web data is usually semi-structured – or heavily unstructured. Secondly, the form of the data in web can be composed of both text and other multimedia type. Although most of the data was in the form of text in past years, multimedia based data like audio and video has become favorite for designing of web sites. Therefore, other than conventional data mining methods which requires mainly structured data, some special techniques should be applied to be able process these two type of different data types simultaneously. In [14], Yong-gui emphasizes the use of "intelligent agents" for the purpose of content mining.

### 2.1.2 Web structure mining

One possible definition for Web structure mining is the process of capturing the most important and meaningful information regarding the overall structure of hyperlinks between web pages as well as the most highlighted or emphasized concepts among pages within the web site. Extracting the meaningful link information via Link Mining [15] is commonly used in link-based classification and clustering to discover the importance of web pages within a site.

On the other hand, there might be some important or core concepts within web pages that best describe the page itself. Web pages are generally composed in the form of HTML or XML mark-up languages in which there exists a predefined page structure. Attributes like HTML tags, bold headings may indicate the importance and provide useful hints. Hence, it would be possible to extract the information from the page by just exploring those specific components having special meanings.

### 2.1.3 Web usage mining

The primary source for web usage mining is the visit logs from users. In other words, the data is produced by the visitors which is not the case with content and structure mining. Web usage mining is defined in [16] as the automatic extraction of clickstream patterns during users' interactions in web sites. In addition, it should be noted that a combination of user specific information together with conceptual – i.e. content – and structural data can be used for Web usage mining.

From a general perspective, within the scope of Web usage mining there are a series of processes performed; data collection, pre-processing, pattern discovery, and pattern analysis. Within the pre-processing stage, the raw data is cleaned – from robot IPs for instance – and classified in terms of users, source IP addresses and time spent on the web site etc. In the pattern discovery, however, possible hidden navigation sequences are highlighted using data mining techniques.. In the analysis phase, the discovered behaviour patterns are processed further as to be an input for recommendation and prediction systems.

## 2.2 Data Sources

Data collection is an important stage in Web mining process. Exploring navigation sequences and hence discovering web users' navigation behaviours require an in depth analysis of a mass volume of data logged. Most commonly, the data to be used is logged on server side via access logs although there are also some other data sources on client side as well. In [16], Mobasher et al. gives a categorization for data sources for Web mining.

### 2.2.1 Usage Data

Server access logs are the primary data source for usage data. Every time a query is initiated on web by users, hits for the requested target HTTP addresses are recorded in log files. The data kept in these log files may include many pieces of information regarding the request such as; IP address, time stamp, the URL accessed, type of the browser, server status, access protocol, resource and possibly some user specific information. More importantly, since the source of request can be distinguished with the originating IP address together with time stamp, entries in access log files can be easily regarded as a base for extracting navigation sequences of specific users. That, in turn, may help much in analyzing users' navigation behaviours leading better prediction systems. A partial log is depicted in Table 2.1.

*Table 2.1.    Partial server access log*

```
66.249.68.87 - [13/Feb/2011:06:39:36 +0200] "GET
/lib/plugins/cow/ical.php?recache=true&dtstart=-1year&exams&/exams.ics
HTTP/1.1" 200 52579 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"

67.195.112.241 - - [13/Feb/2011:06:53:47 +0200] "GET /~e1449271/style.css
HTTP/1.0" 304 - "http://www.ceng.metu.edu.tr/~e1449271/" "Mozilla/5.0
(compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)"

66.249.68.87 - - [13/Feb/2011:06:54:08 +0200] "GET
/index.php?id=undergrad/courses&crsprogram=all&crsyear=20101
HTTP/1.1" 200 4742 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"

66.249.68.47 - [13/Feb/2011:06:39:44 +0200] "GET
/research/modsim/index?printview=0&do=login&sectok=6ffc3fc17c0a14e275f
4b48625368e91 HTTP/1.1" 200 3281 "-" "Mozilla/5.0 (compatible;
Googlebot/2.1; +http://www.google.com/bot.html)"

175.158.29.209 - - [13/Feb/2011:06:56:23 +0200] "GET
/people/faculty/skalkan/index?do=login&sectok=0177bdeab956e HTTP/1.1"
200 10217 "-" "Yeti/1.0 (NHN Corp.; http://help.naver.com/robots/)"
```

### 2.2.2 Content data

Especially for semantic similarity analyses, the content data within web pages and in turn within web sites is crucial and an absolute data source. To enhance the semantic correspondence of web pages and to relate them with predefined taxonomic relations for similarity measures, content data is an inevitable source. The content data is generally in the form of text and multimedia data in XML and/or HTML web pages.

### 2.2.3 Structure data

The mining process also makes use of the structural design of web sites. These structures are usually composed of the hyperlinks between the pages and generally captured via an automated mechanism. Hyperlink structures of web sites are highly utilized as data sources for analysis of users' navigation behaviours as well as for designing better prediction systems.

### 2.2.4 User data

User data is also another important component in terms of data source in data mining process. Past visit logs, purchase histories and comments by the users may include valuable information unique to web users' behaviours and their interests. Furthermore, some demographic data recorded previously – possibly within users' registered profiles – can also be easily integrated.

# CHAPTER 3

# IMPROVING THE PREDICTION OF PAGE ACCESS BY USING SEMANTICALLY ENHANCED CLUSTERING

## 3.1 Preliminaries

### 3.1.1 Defining Ontology

Extracting meaningful information via web-log mining needs essential similarity associations between web pages. For this, there are alternatives that you can choose of, such as, measuring similarities between URL matches or conceptual similarities that exists in the text body of pages. The conceptual similarity is more convenient and proved to provide more strong and meaningful analytics rather than just relying upon URL string matches. In this study, instead of using the text content of web pages only, each web page is associated with a set of keywords – thereafter called *concepts* – that best describes it.

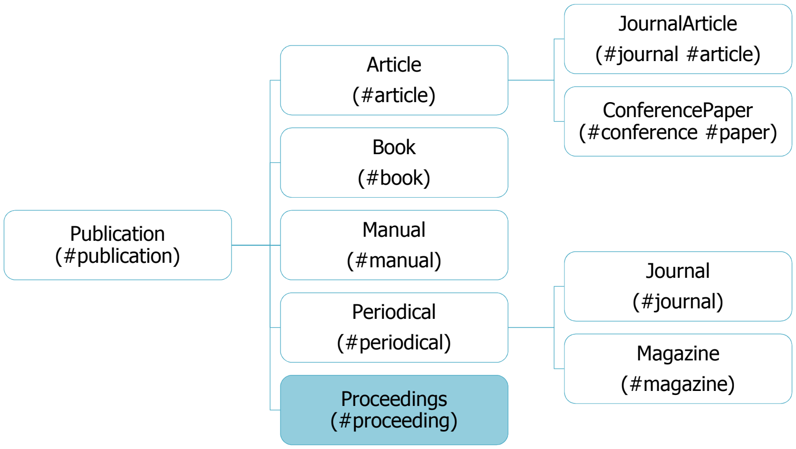


*Figure 3.1.* *Partial ontology*

The similarity between web pages is then measured by considering the similarity between these keywords. As a prerequisite, an ontology of keywords is constructed to reflect the relationships among these concepts. The ontology involving an ISA hierarchical model for this study is based on the Computer Science Department ontology from "Simple HTML Ontology Extensions" project [17]. A sample part of this ontology is presented in Figure 3.1. The original ontology is available in Appendix A at the end of the thesis.

### 3.1.2 Defining Concepts

A crucial part of the study is to define concepts within the ontology together with their associated set of keywords. Each concept is assigned with an "id" and is defined as a set of sets in which keywords are defined. A sample of table is given in Table 3.1.

*Table 3.1. Sample concepts and their respective set of keywords*

| *id* | *Concept* | *Associated keyword sets* |
|---|---|---|
| 4 | ResearchLaborator | {research,laboratory} |
| 11 | Bioinformatics | {bioinformatics} |
| 67 | DataMining | {data, mining} {clustering} |
| 8 | OperatingSystems | {operating,system},{thread},<br>{process},{deadlock} |
| 92 | UndergraduateStudent | {undergraduate,student} |
| 55 | GraduateStudent | {graduate,student} |
| 12 | Course | {course} |
| 40 | GraduateCourse | {graduate, course} |
| 14 | Thesis | {thesis} |
| 7 | Research | {research} |

In other words, a concept set is defined as a set of keyword sets. In particular, the concept set for the concept "*OperatingSystems*" is { {operating,system}, {thread} }.

### 3.1.3 Associating Web Pages with Concepts

In order to label a web page with a concept, all keywords in one of that particular concept's associated keyword set item must appear in the page content. As an example, in order to be able to associate a web page with the concept "*OperatingSystems*" both the words "*operating*" and "*system*", or one of the words "*process*", "*thread*", "*deadlock*" must be with the content. Let the page "www.ceng.metu.edu.tr/research/mining/index" contain "*clustering*" and "*student*". Then the page is labeled with the concept "*DataMining*" but not with "*UndergraduateStudent*" as the later concept requires "*undergraduate*" keyword as well.

As another example, let the web page – /grad/ms – in the dataset has the concept set { Course, GraduateCourse, Thesis, Research }. Referring back to concept definition section above then the *label set* for this web page would be the set of concept ids;

$$L_w = \{\ c_7, c_{12}, c_{14}, c_{40}\ \} \quad or \quad L_w = \ \{\ 7, 12, 14, 40\ \}$$

Hereafter, label set will be used through the implementation in this study rather than referring to *text based* labels.

### 3.1.4 Concept Similarity

The next essential task is to define similarities between the concepts defined in the ontology. Since each web page is associated with concepts, finding similarity between concepts constitute the basics for web page similarity.

The measurement of concept semantic similarity can be accomplished via many methods that can be further divided into four main categories [18];

1. *Edge Counting Methods*: These methods measure the similarity between two concepts $(c_1, c_2)$ by determining the path connecting the concepts in the hierarchical taxonomy with regard of their position

2. *Information Content Methods*: The base parameter for calculating the similarity within these methods is the "*Information content*" of each concept

3. *Feature based Methods*: The features of the concepts are also considered in order to measure similarity

4. *Hybrid methods*: Those methods combine ideas from the above three approaches in order to compute semantic similarity between $(c_1, c_2)$

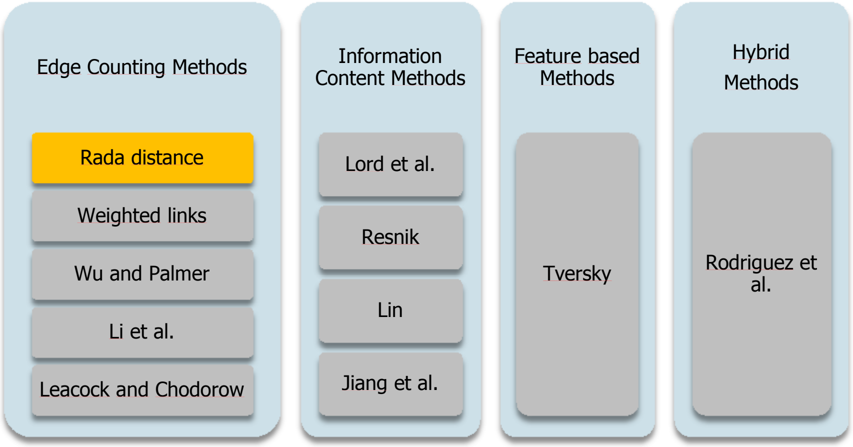


*Figure 3.2.* *Semantic similarity methods*

Some well-known measurement techniques are given in Figure 3.2 and can be further studied in [19,20]. For simplicity, Rada et al.'s Distance [21] is used in this study for measuring the concept similarity. Given two concepts ($c_1$, $c_2$); Rada et al.'s Distance can be defined as the length of the shortest path between $c_1$ and $c_2$ within the hierarchical taxonomic relations framework and given by;

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

Referring to Figure 3.3, let $c_1$=*"research assistant"* and $c_2$=*graduate*, then the distance between $c_1$ and $c_2$ would be simply the length of the path <*research, assistant, worker, person, student, graduate*>. Hence the Rada et al.'s Distance is;

$$dist_{Rada}(c_1, c_2) = 5$$

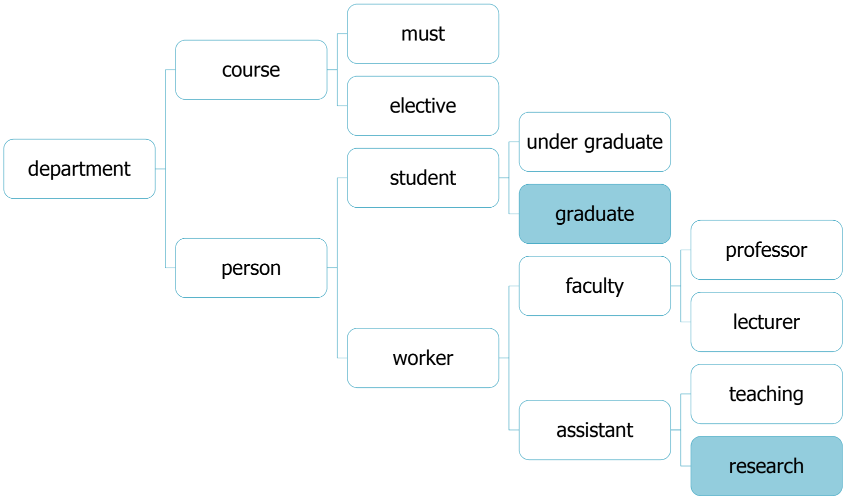


*Figure 3.3.* *Sample concept taxonomy*

### 3.1.5 Concept Set Similarity

To measure the similarity between two sets of concepts, different approaches may be used [20]. In this study, a derivative of Jaccard Similarity approach is used.

### 3.1.6 Modified Jaccard set similarity

Given the similarity matrix for each concept pair, the similarity between two sets are calculated as follows;

- $\forall$ a $\in$ A, $\forall$ b $\in$ B find $max\{sim_{jaccard}(a,b)\}$
- $\forall$ c $\in$ B\A, $\forall$ a $\in$ A find $max\{sim_{jaccard}(c,a)\}$
- find the average of all max similarity values

```
for each ca in set_A:
    max_a = 0
    for each cb in set_B:
        if max_a < conceptSimMatrix[ca][cb]
    max_a = conceptSimMatrix[ca][cb]
    s = s + max_a
set_D = set_B.difference(set_A)
max_a = 0
    for cd in set_D:
        max_a = 0
        for ca in set_A:
            if max_a < conceptSimMatrix[cd][ca]
                max_a = conceptSimMatrix[cd][ca]
        s = s + max_a
    avg = s / len(set_A.union(set_B))
    return avg
```

*Figure 3.4.* *Modified Jaccard similarity*

*Example:*

Let $A = \{x, y, z, k\}$ and $B = \{x, y, z, w\}$. Note that two sets differ with one components in each.

<table>
<tr><th colspan="5"><i>similarity matrix</i></th><th><i>similarity measures</i></th></tr>
<tr><td></td><td><i>x</i></td><td><i>y</i></td><td><i>z</i></td><td><i>w</i></td><td></td></tr>
<tr><td><i>x</i></td><td>1</td><td>0.3</td><td>0.6</td><td>0.1</td><td rowspan="4">∀ a ∈ A similarity values; $\{1.0, 1.0, 1.0, 0.4\}$<br>∀ b ∈ B\A similarity values; $\{0.6\}$</td></tr>
<tr><td><i>y</i></td><td>0.4</td><td>1</td><td>0.5</td><td>0.6</td></tr>
<tr><td><i>z</i></td><td>0.6</td><td>0.8</td><td>1</td><td>0.1</td></tr>
<tr><td><i>k</i></td><td>0.4</td><td>0.2</td><td>0.3</td><td>0.1</td></tr>
</table>

*Figure 3.5. Modified Jaccard example*

Hence the similarity between sets A and B is;

$$sim_{jaccard}(A, B) = \frac{1 + 1 + 1 + 0{,}4 + 0{,}6}{5} = \frac{4}{5} = 0{,}8$$

### 3.1.7 Simple comparison between Average Linkage and Modified Jaccard set similarity

Just to show the effectiveness of the Modified Jaccard similarity function, a simple comparison between our similarity function and one of the well-known set similarity function – i.e. Average Linkage – is provided in this section.

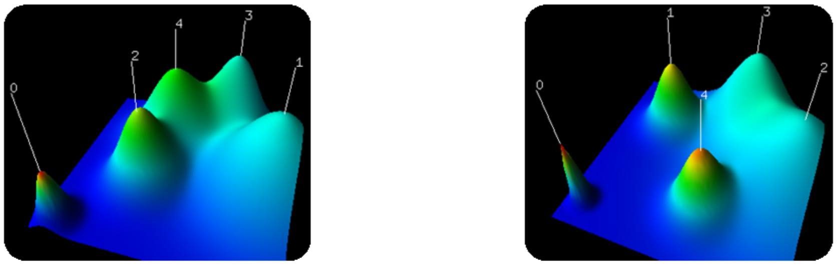
Given the same similarity matrix above, Average Linkage similarity yields;

$$sim_{AVG}(A, B) = \frac{8}{16} = 0{,}5$$

```
for each ca in set_A:
    for each cb in set_B:
        s = s + conceptSimMatrix[ca][cb]
   avg = s / (len(set_) + len(set_B))
   return avg
```

*Figure 3.6.    Average Linkage*

Applying different similarity algorithms yields more effective clustering results, which in turn, helps to identify more accurate clustering schemas even by utilizing exactly the same parameters such as dataset, concept similarity, cluster method, similarity function, intended cluster number and number of iterations. A sample clustering[1] comparison is given in Figure 3.7. The clusters created using our modified Jaccard set similarity approach present more distinct clusters compared to regular Average Linkage based similarity measures.



clusters based on Average Linkage set similarity

clusters based on modified JACCARD set similarity

*Figure 3.7.    Comparison of sample clustering models*

### 3.1.8 Session Similarity

Sessions are defined as a sequence of web pages navigated in a certain

[1] created in gCLUTO – tool developed by Karypis Labs for clustering high volume datasets and analyzing the characteristics of the clusters, http://glaros.dtc.umn.edu/gkhome/cluto/gcluto/overview

predefined time (page stay and total duration) frame. That is to say, a session is composed of a series of web pages.

$$S_i = \{w_{i0}, w_{i1}, w_{i2}, \ldots w_{in}\}$$

With the assumption that each web page is labeled with their appropriate set of concepts in the *labeling* step, a session can be regarded as a collection or a *sequence* of label sets. Hence, $S_i$ can be represented as;

$$seq_i = \left\{L_{w_{i0}}, L_{w_{i1}}, L_{w_{i2}}, \ldots L_{w_{in}}\right\}$$

Measuring the similarity between two sessions, then, is simply the process of aligning their label sequences. For measuring the similarity between two sessions, each label set of each session are treated as two distinct sequences and thereafter the alignment score of these sequences are calculated. Suppose that sessions $S_i$ and $S_j$ have label sets as;

$$seq_i = \{L_{i0}, L_{i1}, L_{i2}, \ldots, L_{in}\} \; where \; S_i \; has \; n \; web \; pages$$

$$seq_j = \{L_{j0}, L_{j1}, L_{j2}, \ldots, L_{jm}\} \; where \; S_j \; has \; m \; web \; pages$$

Hence label sets stand for the sequences on which we can work to align them to quantify the similarity between two sessions. For this, *Needleman-Wunsch* algorithm is used [22].

The Needleman-Wunsch algorithm is one of the best approach promising the optimum alignment independent from the length and/or complexity of sequences [23].

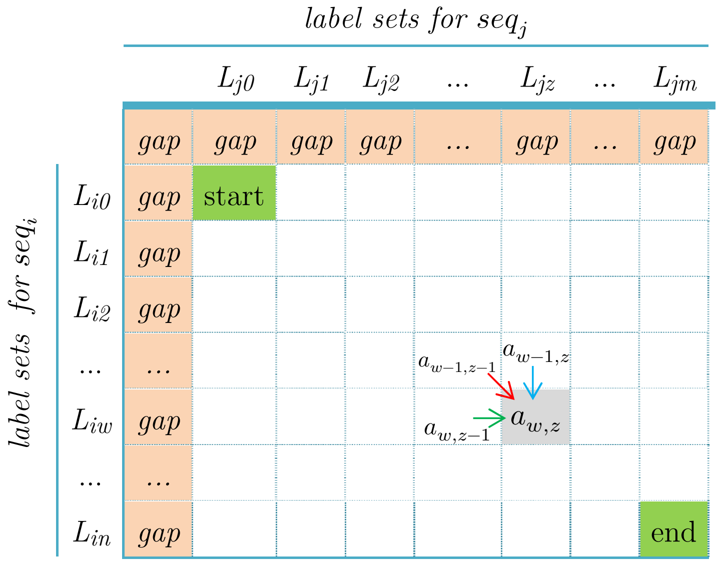


*Figure 3.8. Needleman-Wunsch sequence alignment*

Within the scope of the domain in this study, the algorithm simply establishes "Needleman similarity matrix" by using the concept set similarity values between each label set pairs.

For any $w \in [1..n]$ and $z \in [1..m]$;

$$a[w,z] = max \left\{ \begin{matrix} a[w,z-1] - gap \\ a[w-1,z-1] + sim(L_{iw}, L_{jz}) \\ a[w-1,z] - gap \end{matrix} \right\}$$

where;

- $w \in [1..n]$ and $z \in [1..m]$
- $L_{iw} \in seq_i$ and $L_{jz} \in seq_j$
- *gap* is the penalty value, for dropping one element from one sequence
- $sim(L_{iw}, L_{jz})$ is the concept set similarity between sets $L_{iw}$ and $L_{jz}$

Each specific cell – say $a[w, z]$ – is simply calculated with the *max* value of a triple, namely;

- the upper adjacent cell value i.e. $a[w-1, z]$
- the left adjacent cell value i.e. $a[w, z-1]$
- the sum of upper-left neighbor i.e. $a[w-1, z-1]$ and the similarity value between $L_{iw}$ and $L_{jz}$ i.e. $sim(L_{iw}, L_{jz})$

The construction of the Needleman-Wunsch similarity matrix start from the cell $(L_{i0}, L_{j0})$ and ends with cell $(L_{in}, L_{jm})$ moving from left to right.

The final value for the session similarity between $(seq_i, seq_j)$ is the normalized value of the cell $(L_{in}, L_{jm})$ with the max length of both sequences, i.e.

$$sim_{(seq_i, seq_j)} = \frac{(L_{in}, L_{jm})}{\max(n, m)}$$

Just to illustrate the mentioned algorithm, suppose that we have two sessions $S_0$ and $S_1$ as given in Table 3.2;

*Table 3.2. Sample sessions with concept sets*

| | *web pages navigated* | *labels* |
|---|---|---|
| $S_0$ | / | { 3,7,52,57,58,149 } |
| | /courses/ceng242/ | { 155 } |
| | /courses/ceng242/menu.html | { 21,155 } |
| | /courses/ceng242/ syllabus.html | { 10,12,13,18,21,25,144,155 } |
| $S_1$ | /grad/mswotceng | { 7,12,40,283 } |
| | /grad/courseswtceng | { 7,8,11,14,21,24,33,214,256 } |
| | /index.php | { 3,7,18,34,36,52,54,57,58,199 } |

*Table 3.2. (continued)*

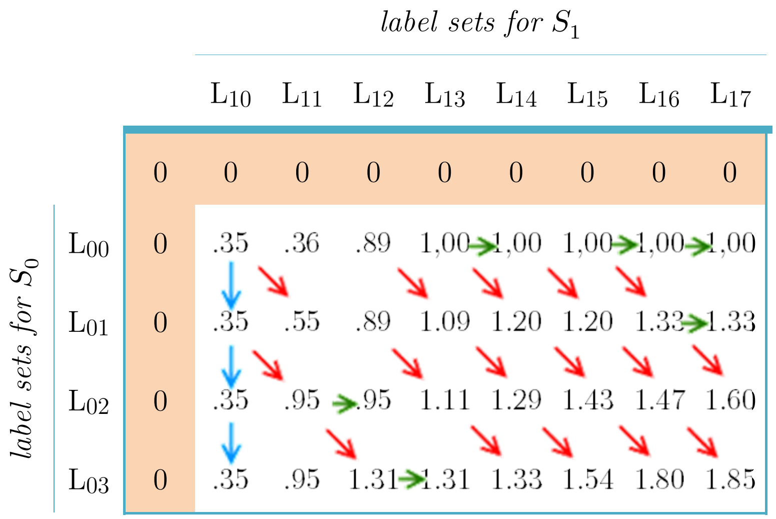
| | *web pages navigated* | *labels* |
|---|---|---|
| $S_1$ | / | { 6,3,7,52,57,58,149 } |
| | /news/20101/pre_evaluation | { 7,57 } |
| | / | { 6,3,7,52,57,58,149 } |
| | /grad/ms | { 7,12,14,40 } |
| | /grad/courses | { 7,14,18,24,28,34,36,52,137,173,241 } |

In addition, assume that the set similarity between label sets are calculated using the Modified-Jaccard as shown in Table 3.3.

*Table 3.3. Modified-Jaccard set similarities*

| | | *label sets for $S_1$* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $L_{10}$ | $L_{11}$ | $L_{12}$ | $L_{13}$ | $L_{14}$ | $L_{15}$ | $L_{16}$ | $L_{17}$ |
| *label sets for $S_0$* | $L_{00}$ | .35 | .36 | .89 | 1,00 | .51 | 1,00 | .36 | .53 |
| | $L_{01}$ | .33 | .20 | .20 | .20 | .20 | .20 | .33 | .20 |
| | $L_{02}$ | .27 | .60 | .27 | .23 | .20 | .23 | .27 | .27 |
| | $L_{03}$ | .33 | .38 | .36 | .25 | .22 | .25 | .37 | .38 |

For simplicity *gap* is assumed to be zero. Then the Needleman Wunsch similarity matrix can be constructed as given in Figure 3.9.



| | | L10 | L11 | L12 | L13 | L14 | L15 | L16 | L17 |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| L00 | 0 | .35 | .36 | .89 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| L01 | 0 | .35 | .55 | .89 | 1.09 | 1.20 | 1.20 | 1.33 | 1.33 |
| L02 | 0 | .35 | .95 | .95 | 1.11 | 1.29 | 1.43 | 1.47 | 1.60 |
| L03 | 0 | .35 | .95 | 1.31 | 1.31 | 1.33 | 1.54 | 1.80 | 1.85 |

*Figure 3.9. Aligned session similarity between $S_0$ and $S_1$*

Hence the session similarity score is;

$$sim_{(seq_0,seq_1)} = \frac{(L_{03}, L_{17})}{\max(3,8)} = \frac{1.85}{8} = 0.23$$

## 3.2 Method

In this study, in order to predict the next web page of a user session, four different approaches have been investigated in two dimensions. In the first dimension two alternatives are explored; namely using direct URL addresses vs using concept sets for representing a web page. In the second dimension, search of the similar sessions of the session whose next page is going to be predicted has been done in two different forms; either by comparing it with all previously captured sessions, or first by choosing the most similar cluster of the sessions and then, by comparing it only with the sessions in that cluster.

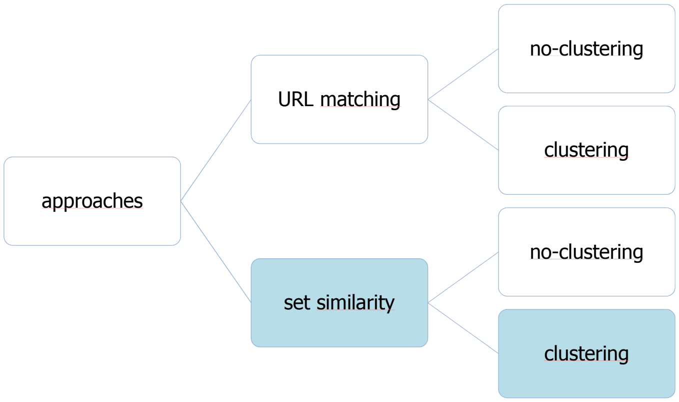


*Figure 3.10. Possible approaches*

As a result, as shown in Figure 3.10, the following four different approaches have been examined:

- Using the session similarity method explained in the previous section the most similar session of the session whose next page is to be predicted is searched within the whole previously recorded session set. The web page accesses of the sessions are represented with their URLs, and therefore the matching between two web pages is either 0 or 1 representing as different or same respectively. This approach is the most natural and naive method.

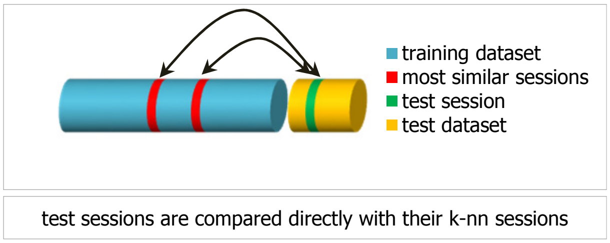


*Figure 3.11. No-clustering option*

- In the second approach still web page accesses are represented with

URLs. However, previously collected sessions, i.e., the session dataset, are clustered according to session similarities described in the previous section. For each cluster, the cluster centroid has also been calculated. Therefore, rather than searching similar sessions of the session whose next web page is going to be predicted in the whole dataset, first the most similar cluster is determined by only comparing it with the centroids of the clusters. After that, all the sessions of that cluster are compared against the current session in order to find the most similar ones. This significantly reduces the number of comparisons among the sessions. However, it potentially misses possible most similar sessions that may be placed into another cluster whose centroid is less similar to the current session than the centroid of the cluster being searched. One aim of this study is to show that the gain in execution time is worth risking the missing most similar sessions.

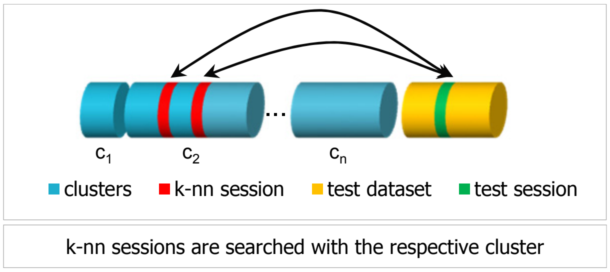


*Figure 3.12. Clustering approach*

- In the third approach rather than representing web pages with their URLs, they have been represented with a set of concepts that are captured from the web pages. The main motivation behind this representation is to be able to capture user's *intention* rather than her recorded behaviour. The web pages with similar concepts sets potentially contain similar information and therefore they are similar too. Therefore, users may choose to view either one of them if they are interested in that kind of information. Using this approach, 0-1 matching between URL names can be relaxed to a fractional matching between two web pages depending on how similar their contents are, which is represented by concept sets. This will affect the session similarity as well, and potentially we will be able to make better estimation among the sessions, and this will lead to better prediction of the next page. In this approach, without clustering the whole dataset of sessions are compared against the session whose next page is going to be predicted. We expect the best result from this approach.

- In the final method, in addition to using concept sets, the session dataset has also been clustered similar to the second approach, and the similar sessions are searched inside the cluster whose centroid is most similar to the session whose next page is to be predicted. As in approach 2, in this approach, we expect the gain in time without losing much on the accuracy result.

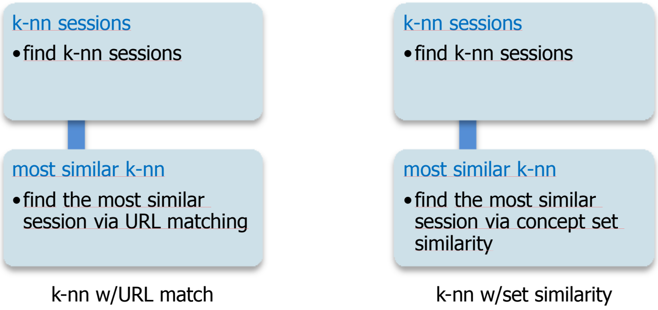


*Figure 3.13.* *URL match vs. set similarity*

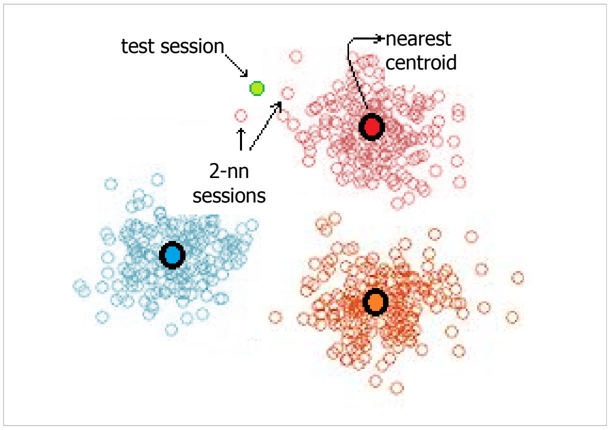


*Figure 3.14. Sample clustering with 3-cluster with 2-nn sessions*

Two out of four methods described above require clustering of the sessions, which we call as "*model construction phase*". The "*prediction phase*" of these methods is also different from the others. When "clustering" is used, the model construction phase contains the following steps:

- All sessions in the dataset are clustered using session similarity method,
- Then, for each cluster the centroid is also determined.

For these methods, the prediction phase also contains two steps:

- For the session whose next page is going to be predicted, the most similar centroid has been determined by comparing it with all the cluster centroids.
- It is compared against all the sessions of the cluster with most similar centroid, and top "*k*" most similar sessions are determined.

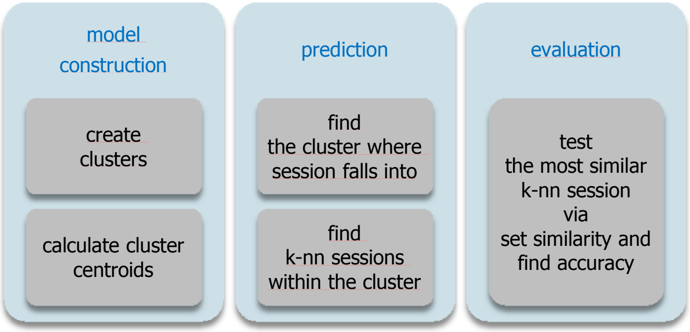

model construction
create clusters
calculate cluster centroids
prediction
find the cluster where session falls into
find k-nn sessions within the cluster
evaluation
test the most similar k-nn session via set similarity and find accuracy

*Figure 3.15.* *Process flow for cluster based recommendation*

When there is no clustering, directly prediction phase is used, and the current session is compared against all the sessions in the dataset.

The first thesis of this work is that using concept sets instead of direct URLs will increase the chance of finding similar sessions and therefore with this approach the accuracy of predicting the next web page to be accessed increases as well. Furthermore, secondly, we claim that clustering previously recorded sessions and searching the most similar session of the current session in a cluster with most similar centroid reduces the whole search time while having a very small and acceptable drop in the accuracy of the prediction. Therefore, below we give the details of the method construction and prediction phases for the final approach that has been introduced above; namely web page accesses are represented by their corresponding concept sets, and the dataset of the previously recorded sessions are clustered by using session similarity method.

## 3.3 Model Construction

We are going to explain the steps of this phase with a small sample of our real life example dataset which is given in Table 3.4. The second column actually corresponds the real session IDs of our dataset, and we have introduced the new IDs (virtual ID) in column three for simplification.

*Table 3.4. Sample dataset*

| sessions | | |
|---|---|---|
| session | real id | virtual id |
| $S_{0}$ | 345 | 0 |
| $S_{1}$ | 466 | 1 |
| $S_{2}$ | 489 | 2 |
| $S_{3}$ | 1142 | 3 |
| $S_{4}$ | 1352 | 4 |
| $S_{5}$ | 1822 | 5 |
| $S_{6}$ | 3168 | 6 |
| $S_{7}$ | 3234 | 7 |
| $S_{8}$ | 3270 | 8 |
| $S_{9}$ | 3631 | 9 |
| $S_{10}$ | 3956 | 10 |
| $S_{11}$ | 4685 | 11 |
| $S_{12}$ | 4794 | 12 |
| $S_{13}$ | 4966 | 13 |
| $S_{14}$ | 5014 | 14 |
| $S_{15}$ | 8040 | 15 |
| $S_{16}$ | 8108 | 16 |
| $S_{17}$ | 8992 | 17 |
| $S_{18}$ | 10722 | 18 |
| $S_{19}$ | 10882 | 19 |
| $S_{20}$ | 11008 | 20 |
| $S_{21}$ | 11468 | 21 |
| $S_{22}$ | 11748 | 22 |
| $S_{23}$ | 11809 | 23 |
| $S_{24}$ | 11990 | 24 |
| $S_{25}$ | 12003 | 25 |
| $S_{26}$ | 13029 | 26 |
| $S_{27}$ | 14501 | 27 |
| $S_{28}$ | 15901 | 28 |
| $S_{29}$ | 17915 | 29 |

The same sample dataset will also be used for evaluating the constructed model in the evaluation part in this study. Therefore, assume that only a random three quarters of the sample dataset is used for model construction. The remaining quarter will be used in evaluation section and regarded as the "test dataset". Suppose that the datasets are set as follows;

model dataset = { 0, 1, 2, 3, 4, 5, 6, 7, 9, 11, 13, 14, 17, 18, 19, 20, 21, 22, 23,
24, 25, 27, 28, 29 }

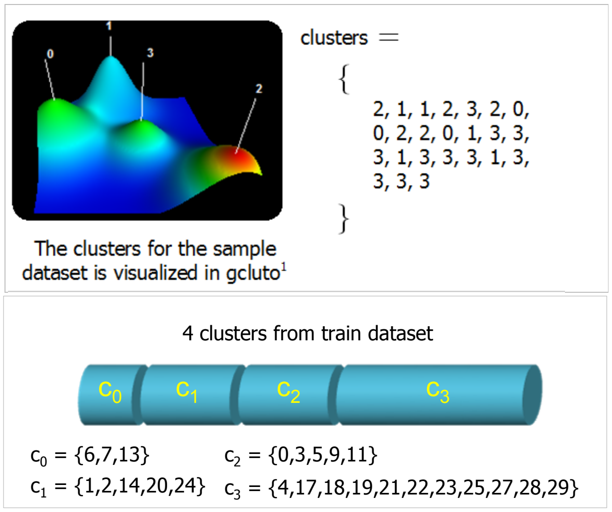
test dataset = { 8, 10, 12, 15, 16, 26 }

Note that upcoming steps will refer to virtual sessions IDs.

### 3.3.1 Cluster the dataset – step 1 of 2

For ease of use, we used *scluster* module within CLUTO package[2]. CLUTO is a commonly used tool for clustering multi-dimensional datasets and for cross-checking the various trends within the clusters.

CLUTO partitioned this model dataset into four clusters and the resulting cluster set for 24-session dataset is given in Figure 3.16.



clusters =
{
2, 1, 1, 2, 3, 2, 0,
0, 2, 2, 0, 1, 3, 3,
3, 1, 3, 3, 3, 1, 3,
3, 3, 3
}

The clusters for the sample dataset is visualized in gcluto[1]

4 clusters from train dataset

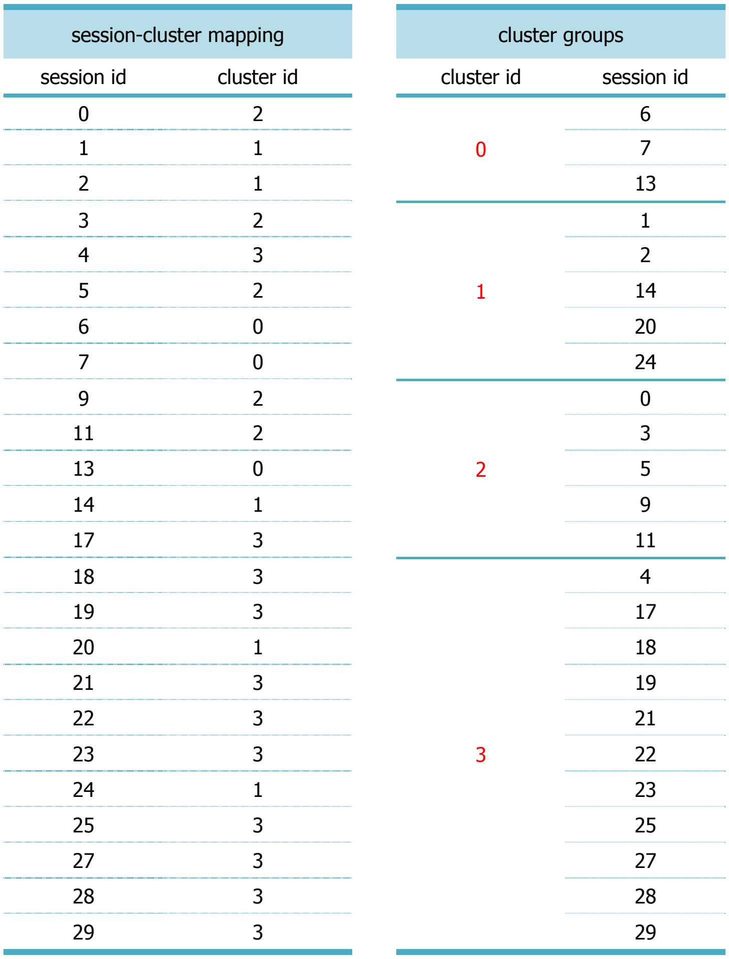
$c_0$ = {6,7,13} $c_2$ = {0,3,5,9,11}
$c_1$ = {1,2,14,20,24} $c_3$ = {4,17,18,19,21,22,23,25,27,28,29}

*Figure 3.16. Test dataset clusters*

Note that each session in the dataset has the corresponding cluster id with the same index in cluster list – *clusters*.

[2] created in *scluster* – http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

*Table 3.5.     Cluster-session distribution*



| session-cluster mapping | | cluster groups | |
|---|---|---|---|
| session id | cluster id | cluster id | session id |
| 0 | 2 | 0 | 6 |
| 1 | 1 | | 7 |
| 2 | 1 | | 13 |
| 3 | 2 | 1 | 1 |
| 4 | 3 | | 2 |
| 5 | 2 | | 14 |
| 6 | 0 | | 20 |
| 7 | 0 | | 24 |
| 9 | 2 | 2 | 0 |
| 11 | 2 | | 3 |
| 13 | 0 | | 5 |
| 14 | 1 | | 9 |
| 17 | 3 | | 11 |
| 18 | 3 | 3 | 4 |
| 19 | 3 | | 17 |
| 20 | 1 | | 18 |
| 21 | 3 | | 19 |
| 22 | 3 | | 21 |
| 23 | 3 | | 22 |
| 24 | 1 | | 23 |
| 25 | 3 | | 25 |
| 27 | 3 | | 27 |
| 28 | 3 | | 28 |
| 29 | 3 | | 29 |

### 3.3.2 Calculate cluster centroids – step 2 of 2

The next step is to find out centroids for each respective cluster. Although there are many, two most convenient methods for calculating centroids are;

- maximum sum of Euclidean similarity between sessions

- maximum sum of squares of Euclidean similarity between sessions

We used the first approach to calculate possible centroid candidates.

$$s(S_i) = \sum_{j=1}^{n} Sim(S_i, S_j)$$

where $S_i, S_j$ are sessions in the cluster, $n$ is the number of sessions in cluster and $j \neq i$. The algorithm is given in Figure 3.17. Assume that session similarity values for the given dataset are already computed in advance as given in Appendix B.

```
centroids = [ ]
for i in range(CLUSTERNUM):

  #construct a list of sessions for each cluster
  currentcluster = [ ]
  for j in range(len(trains_clusters)):
    if i == trains_clusters[j]:
      currentcluster.append(trains[j])

  #intercluster similarity calculation
  sim_sum_list = [ ]
  for a in currentcluster:
    sim_sum = 0
    for b in currentcluster:
      if a <> b:
        sim_sum += sims[a][b]
    sim_sum_list.append((a,sim_sum))

  sim_sum_list.sort(key=operator.itemgetter(1))
  sim_sum_list.reverse()

  centroids.append((i,sim_sum_list[0][0]))
```

*Figure 3.17. Cluster centroid calculation*

To find cluster centroids for each cluster, we test all sessions within that cluster by calculating the sum of their respective inter-cluster similarities based on the session similarity values.

### Centroid for cluster #0

For cluster #0, we test all sessions in that cluster, namely { 6, 7, 13 }, against the rest by calculating the sum of their respective inter-cluster similarities. The table below summarizes similarity values together with their Euclidian sum for cluster #0 sessions;

|  | 6 | 7 | 13 | ∑ |
|---|---|---|---|---|
| 6 | - | .80 | .80 | 1.60 |
| 7 | .80 | - | .51 | 1.31 |
| 13 | .80 | .51 | - | 1.31 |

*Figure 3.18. Inter-cluster session similarity matrix for cluster#0*

Session #6 has the maximum sum of session similarity value, hence the centroid for the cluster #0 is session #6.

### Centroid for cluster #1

Session #14 has the maximum sum of session similarity value, hence the centroid for the cluster #1 is session #14.

|  | 1 | 2 | 14 | 20 | 24 | ∑ |
|---|---|---|---|---|---|---|
| 1 | - | .68 | .60 | .39 | .60 | 2,27 |
| 2 | .68 | - | .69 | .44 | .80 | 2,61 |
| 14 | .60 | .69 | - | .41 | 1,00 | 2,70 |
| 20 | .39 | .44 | .41 | - | .29 | 1,54 |
| 24 | .60 | .80 | 1,00 | .29 | - | 2,69 |

*Figure 3.19. Inter-cluster session similarity matrix for cluster#1*

### Centroid for cluster #2

Session #11 has the maximum sum of session similarity value, hence the centroid for the cluster #2 is session #11.

|  | 0 | 3 | 5 | 9 | 11 | Σ |
|---|---|---|---|---|---|---|
| 0 | - | .18 | .40 | .24 | .22 | 1,04 |
| 3 | .18 | - | .37 | .39 | .59 | 1,53 |
| 5 | .40 | .37 | - | .55 | .48 | 1,80 |
| 9 | .24 | .39 | .55 | - | .59 | 1,77 |
| 11 | .22 | .59 | .48 | .59 | - | 1,88 |

*Figure 3.20. Inter-cluster session similarity matrix for cluster#2*

### Centroid for cluster #3

Session #28 has the maximum sum of session similarity value, hence the centroid for the cluster #3 is session #28.

|  | 4 | 17 | 18 | 19 | 21 | 22 | 23 | 25 | 27 | 28 | 29 | Σ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | - | .57 | .55 | .67 | .60 | .53 | .45 | .61 | .55 | .66 | .75 | 5,94 |
| 17 | .57 | - | .80 | .52 | .48 | .41 | .40 | .56 | .60 | .75 | .52 | 5,61 |
| 18 | .55 | .80 | - | .54 | .53 | .36 | .58 | .51 | .80 | .62 | .45 | 5,72 |
| 19 | .67 | .52 | .54 | - | .52 | .54 | .43 | .58 | .44 | .55 | .42 | 5,21 |
| 21 | .60 | .48 | .53 | .52 | - | .44 | .47 | .53 | .53 | .55 | .67 | 5,32 |
| 22 | .53 | .41 | .36 | .54 | .44 | - | .32 | .47 | .36 | .41 | .42 | 4,26 |
| 23 | .45 | .40 | .58 | .43 | .47 | .32 | - | .55 | .85 | .71 | .47 | 5,22 |
| 25 | .61 | .56 | .51 | .58 | .53 | .47 | .55 | - | .42 | .53 | .44 | 5,20 |
| 27 | .55 | .60 | .80 | .44 | .53 | .36 | .85 | .42 | - | .70 | .48 | 5,72 |
| 28 | .66 | .75 | .62 | .55 | .55 | .41 | .71 | .53 | .70 | - | .63 | 6,11 |
| 29 | .75 | .52 | .45 | .42 | .67 | .42 | .47 | .44 | .48 | .63 | - | 5,24 |

*Figure 3.21. Inter-cluster session similarity matrix for cluster#3*

The resulting set of cluster centroids is;

$$\begin{aligned} \text{centroids} &= \{\ (c_i,\, s_j) \mid c_i \text{ cluster id and } s_j \text{ session id }\} \\ &= \{\ (0,\,6),\ (1,\,14),\ (2,\,11),\ (3,\,28)\ \} \end{aligned}$$

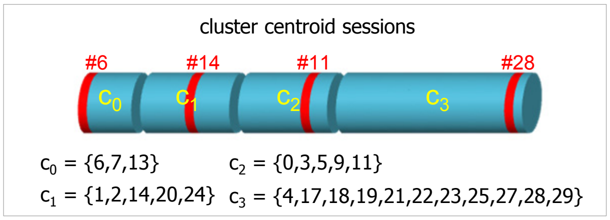
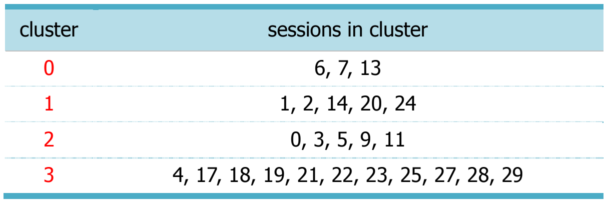


*Figure 3.22. Centroids*

*Table 3.6. Session distribution among clusters*



| cluster | sessions in cluster |
|---|---|
| 0 | 6, 7, 13 |
| 1 | 1, 2, 14, 20, 24 |
| 2 | 0, 3, 5, 9, 11 |
| 3 | 4, 17, 18, 19, 21, 22, 23, 25, 27, 28, 29 |

## 3.4 Prediction

### 3.4.1 Find the cluster for the session – step 1 of 2

To accomplish this step, each session is compared against each centroid session one-by-one in order to determine which centroid session is the most similar to the current test session.

Recall that the test dataset session list is; test = { 8, 10, 12, 15, 16, 26 }. Assume that we are trying to make a prediction for the session #15. Let the following similarity matrix as shown in Table 3.7 stands for the similarity values between session #15 and all cluster centroid sessions, i.e. centroids = { (0, 6), (1, 14), (2, 11), (3, 28) };

*Table 3.7. Cluster selection for session #15*

| *similarity matrix*<br>*between test session #15 and cluster centroid sessions* | | | | |
|---|---|---|---|---|
| cluster id | 0 | 1 | 2 | 3 |
| centroid session id | 6 | 14 | 11 | 28 |
| similarity | .50 | .80 | .18 | .38 |

Since the test session #15 is most similar with cluster #1 centroid session #14, it falls into cluster #1.

```
for test in tests:
# find the most similar centroid, hence its cluster
    clusterid = 0
    max_sim = -1
# (clusterid, sessionid)
    for a,b in centroids:
        if max_sim < sims[b][test]:
            max_sim = sims[b][test]
            clusterid = a
            centroidid = b

        print 'test session %d falls in cluster %d with max sim %f with
          centroid session %d' % (test, clusterid, max_sim, centroidid)
```

*Figure 3.23. Finding the cluster*

Similarly, for the other test dataset sessions, the same procedure yields the clusters that each test session belongs to. Below is the list of clusters for which the rest of the test sessions fall into;

**cluster for test session #10**

| | | | | |
|---|---|---|---|---|
| cluster id | 0 | 1 | 2 | 3 |
| centroid session id | 6 | 14 | 11 | 28 |
| similarity | .11 | .12 | .49 | .36 |

**cluster for test session #12**

| | | | | |
|---|---|---|---|---|
| cluster id | 0 | 1 | 2 | 3 |
| centroid session id | 6 | 14 | 11 | 28 |
| similarity | .75 | .18 | .23 | .28 |

**cluster for test session #8**

| | | | | |
|---|---|---|---|---|
| cluster id | 0 | 1 | 2 | 3 |
| centroid session id | 6 | 14 | 11 | 28 |
| similarity | .33 | .26 | .51 | .67 |

**cluster for test session #16**

| | | | | |
|---|---|---|---|---|
| cluster id | 0 | 1 | 2 | 3 |
| centroid session id | 6 | 14 | 11 | 28 |
| similarity | .22 | .18 | .51 | .74 |

**cluster for test session #26**

| | | | | |
|---|---|---|---|---|
| cluster id | 0 | 1 | 2 | 3 |
| centroid session id | 6 | 14 | 11 | 28 |
| similarity | .33 | .80 | .13 | .38 |

Figure 3.24 summarizes distribution of test sessions among clusters;

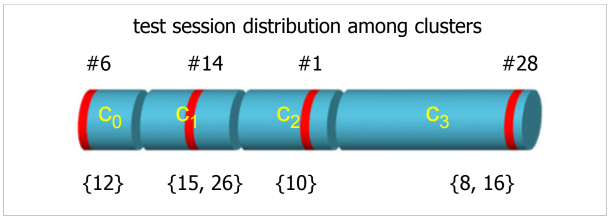


*Figure 3.24. Distribution of test sessions among clusters*

### 3.4.2 Find *k-nn* sessions for the session within its respective cluster – step 2 of 2

Upon finding clusters in which test sessions fall into, another search is then initiated – within the respective cluster – for looking the *k-nn* sessions for each of the test session.

```
for test in tests:
    # get all neighbors in train
    dists = [ ]
    for train in trains:
        dists.append( (train, sims[train][test]) )
    dists.sort(key=operator.itemgetter(1))
    dists.reverse()

    # get K nearest neighbors
    knn = [ ]
    for i in range(K):
        knn.append (dists[i][0])
```

*Figure 3.25. Finding k-nn sessions*

The purpose is to find the most similar session(s) for each test session within its cluster so that it would be more faster to measure set similarities between test sessions and their respective *k-nn* sessions.

For the illustration, we used *2-nn* and tried to find two sessions which are most similar to the session for which we are trying to make the prediction. By simply utilizing the session similarity matrix; we have the following; as the session #15 falls into cluster #1, sessions in cluster #1 are to be considered for finding its respective *k-nn* sessions. The cluster #1 has the following sessions; { 1, 2, 14, 20, 24 }. Table 3.8 gives for the similarity matrix between test session #15 and the sessions of the cluster that it falls into.

*Table 3.8. k-nn sessions for session #15*

|  | 1 | 2 | 14 | 20 | 24 |
|---|---|---|---|---|---|
| 15 | .68 | 1.0 | .80 | .49 | .80 |

For the test session #15, the set of *2-nn* sessions together with similarity values – { ($s_i$, $sim_i$) | where $s_i$ is the session id in cluster #1 and $sim_i$ is the similarity between $s_i$ and the test session #15 } – is { (2, 1.00), (24, 0.80) }. Hence, *2-nn* sessions for the session #15 in cluster #1 are { 2, 24 }.

*Table 3.9. Pages navigated in session #15*

|  | *web pages navigated* | *labels* |
|---|---|---|
| $S_{15}$ | /courses/ceng242 | { 21, 155 } |
|  | /courses/ceng242/syllabus.html | { 10, 12, 13, 18, 21, 25, 144, 155} |
|  | /courses/ceng242/assignments/ | { 21, 155 } |

*Table 3.10. Pages navigated in most similar k-nn session*

|  | *web pages navigated* | *labels* |
|---|---|---|
| $S_2$ | /courses/ceng242/ | { 21, 155 } |
|  | /courses/ceng242/assignments/ | { 21, 155 } |
|  | /courses/ceng242/assignments/2009 | { 21, 155 } |
|  | /courses/ceng242/documents/ | { 18, 21, 155 } |

In the sample dataset, navigation sequences and corresponding concept labels for session #15 and its most similar *k-nn* session #2 are given in Table 3.9 and

Table 3.10 respectively.

Hence, the prediction for the next navigation for session #15 would be the last URL of the session #2, i.e. "*/courses/ceng242/documents/*".

Similarly, we can evaluate most similar *k-nn* sessions for the rest of the test dataset sessions simply by utilizing the master session similarity matrix;

**Session(s) falling in cluster #0 :**

Session #12 is in cluster #0 and cluster #0 has the following sessions; { 6, 7, 13}. Table 3.11 is the similarity matrix between test session #12 and the sessions of the cluster that it falls into.

*Table 3.11. k-nn sessions for session #12*

| | 6 | 7 | 13 |
|---|---|---|---|
| 12 | .75 | .65 | .20 |

The list of *k-nn* sessions – $(s_i,\ sim_i)$ where $s_i$ is the session id in cluster #0 and $sim_i$ is the similarity between test session #12 and $s_i$. – are; { (6, 0.75), (7, 0.650715), (13, 0.197405) }. Hence, the 2-nn sessions in cluster #0 for the test session #12 are { 6, 7 }.

**Session(s) falling in cluster #1 :**

Apart from session #15, session #26 also falls into cluster #1. The cluster #1 has the following sessions; { 1, 2, 14, 20, 24 }. Table 3.12 is for the similarity matrix between test session #26 and the sessions of the cluster that it falls into.

*Table 3.12. k-nn sessions for session #26*

|  | 1 | 2 | 14 | 20 | 24 |
|---|---|---|---|---|---|
| 26 | .62 | .75 | .80 | .29 | .54 |

2-nn sessions in cluster #1 for the test session #26 are { 14, 2 }.

**Session(s) falling in cluster #2:**

Test session #10 falls into cluster #2. The cluster #2 has the following sessions; { 0, 3, 5, 9, 11 }. *k-nn* sessions in cluster #2 for the test session #10 are { (5, 0.519), (11, 0.493745), (9, 0.477502), (3, 0.410889), (0, 0.311113) }. Hence, the 2-nn sessions for test session #10 are { 5, 11 }.

*Table 3.13. k-nn sessions for session #10*

|  | 0 | 3 | 5 | 9 | 11 |
|---|---|---|---|---|---|
| 10 | .31 | .41 | .52 | .48 | .49 |

**Session(s) falling in cluster #3 :**

Similarly, test sessions { 8, 16 } fall into cluster #3, therefore sessions in cluster #3 are to be considered for finding *k-nn* sessions. The cluster #3 has the following sessions; { 4, 17, 18, 19, 21, 22, 23, 25, 27, 28, 29 }. Table 3.14 is for the similarity matrix between test sessions { 8, 16 } and the sessions of the cluster that they fall into.

*Table 3.14. k-nn sessions for session #8 and #16*

|  | 4 | 17 | 18 | 19 | 21 | 22 | 23 | 25 | 27 | 28 | 29 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | .52 | .56 | .76 | .43 | .57 | .42 | .61 | .44 | .83 | .67 | .64 |
| 16 | .60 | .73 | .66 | .89 | .52 | .40 | .58 | .66 | .59 | .74 | .52 |

In cluster #3, *2-nn* sessions for test sessions #8 and #16 are { 27, 18 } and

{ 19, 28 } respectively. Table 3.15 summarizes all *2-nn* sessions.

*Table 3.15. k-nn sessions for each test session*

| *test dataset* | | |
|---|---|---|
| *session* | *cluster* | *2-nn sessions* |
| 8 | 3 | { 27, 18 } |
| 10 | 2 | { 5, 11 } |
| 12 | 0 | { 6, 7 } |
| 15 | 1 | { 2, 24 } |
| 16 | 3 | { 19, 28 } |
| 26 | 1 | { 14, 2 } |

# CHAPTER 4

# EVALUATION

In this section, we introduce the dataset, the accuracy measure and the comparison of the methods introduced above for the prediction of the next web page to be visited by the web user.

## 4.1 Dataset

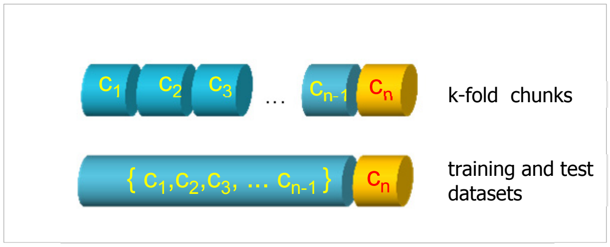
A real dataset obtained from web logs[3] of Computer Engineering Department (CENG) at METU have been used for the evaluation. *k-fold* cross validation technique has been utilized to measure the accuracy of the next page prediction methods introduced above. The dataset belongs to a year (2012) of access web-logs of METU CENG website (http://www.ceng.metu.edu.tr) and contains 293,969 access log items in raw mode. After cleaning phase 33,690 log items remained. The data is on Apache HTTP server combined log format. The web site has 4,371 distinct URLs and 3,538 unique IP addresses. The total number of concepts defined on the website is 301 and the total number of sessions is 1,126. Average number of concept for a webpage is 2.87 (max 45).

### k-fold chunks

Dataset is split into *k-fold* chunks at the beginning and at every iteration of *k* rounds, one distinct chunk is assigned as the *test* dataset and remaining chunks

[3] year of access web-logs of METU C.Eng. website (http://www.ceng.metu.edu.tr)

are merged to form a single *training* dataset. *5-fold* cross validation is used for evaluation.



$C_1$ $C_2$ $C_3$ ... $C_{n-1}$ $C_n$
k-fold chunks
$\{C_1, C_2, C_3, \ldots C_{n-1}\}$ $C_n$
training and test datasets

```
def chunks(l, n):
   return [l[i:i+n] for i in range(0, len(l), n)]

def split_dataset(dataset):
   dataset_copy = dataset[:]
   random.shuffle(dataset_copy)
      return chunks(dataset_copy,
      int(math.ceil(float(len(dataset)) / float(KFOLD))))
   :::
# split dataset in <k-fold> partitions
chunkset = split_dataset(range(len(sessions)))
```

*Figure 4.1. Creating chunks*

## 4.2 Measure accuracy rates

Since our methods that try to predict the next page to be visited generate $k$ potential web pages, we measure the accuracy in terms of the similarity between these predicted pages and the actual page visited by the web user. We take the highest similarity value as the accuracy value.

Last URLs of sessions are used for calculating the set similarity. That is, the last URL of the test session and the most similar sessions (k nearest neighbors) determined by the proposed prediction methods are compared with each other using concept set similarity method. Concept sets which are pre-labeled with each URL are used in calculating the concept set similarity value as explained in the previous section.

Referring back to the test dataset, a sample illustration is the concept set similarity between the test session #10 and its 2-knn sessions #5 and #11. The set of concepts labeled for the last URLs of sessions #10 and #11 are as follows respectively;

$$URL_{10} = \{\ 3,\ 7,\ 8,\ 34,\ 36,\ 57,\ 241\ \}$$

$$URL_{11} = \{\ 7,\ 21,\ 27,\ 33,\ 34,\ 39,\ 214,\ 241\ \}$$

```
settest = set()   # label set for last URL of test session
settest = settest.union(set(sessions[test].pageviews[-1][0].labels))

max_test = 0
for s in knn:
    setknn = set()   # label set  for k-nn session
    setknn = setknn.union(sessions[s].pageviews[-1][0].labels)
    avg = pageviewsim.JaccardSim(settest, setknn, concepts,
       conceptSimMatrix)
    if avg > max_test:
        max_test = avg

sum_avg += max_test
return sum_avg / len(tests)
```

*Figure 4.2.* *Concept set similarity calculation*

The concept set similarity matrix for those concepts labeled for the last URLs of sessions #10 and #11 is;

| | | concept set for $URL_{11}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 21 | 27 | 33 | 34 | 39 | 214 | 241 |
| concept set for $URL_{10}$ | 3 | .25 | .25 | .25 | .25 | .25 | .20 | .17 | .14 |
| | 7 | 1,00 | .20 | .20 | .20 | .20 | .25 | .33 | .13 |
| | 8 | .20 | .33 | .33 | .33 | .33 | .17 | .14 | .13 |
| | 34 | .20 | .33 | .33 | .33 | 1,00 | .17 | .14 | .13 |
| | 36 | .20 | .33 | .33 | .33 | .33 | .17 | .14 | .13 |
| | 57 | .17 | .17 | .17 | .17 | .17 | .14 | .13 | .11 |
| | 241 | .13 | .13 | .13 | .13 | .13 | .11 | .10 | 1,00 |

*Figure 4.3. Concept set similarity between sessions #10 and #11*

$$sim_{jaccard}(URL_{10}, URL_{11}) = \frac{5{,}67}{12} = 0{,}47$$

Similarly we can calculate the concept set similarity between test session #10 and its next *k-nn* session – i.e. session #5.

| | | concept set for $URL_5$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3 | 7 | 18 | 34 | 36 | 52 | 54 | 57 | 58 | 199 |
| concept set for $URL_{10}$ | 3 | 1,00 | .25 | .25 | .25 | .25 | .20 | .20 | .20 | .20 | .17 |
| | 7 | .25 | 1,00 | .20 | .20 | .20 | .17 | .50 | .17 | .17 | .33 |
| | 8 | .25 | .20 | .33 | .33 | .33 | .17 | .17 | .17 | .17 | .14 |
| | 34 | .25 | .20 | .33 | 1,00 | .33 | .17 | .17 | .17 | .17 | .14 |
| | 36 | .25 | .20 | .33 | .33 | 1,00 | .17 | .17 | .17 | .17 | .14 |
| | 57 | .20 | .17 | .17 | .17 | .17 | .33 | .14 | 1,00 | .14 | .13 |
| | 241 | .14 | .13 | .13 | .13 | .13 | .11 | .11 | .11 | .11 | .10 |

*Figure 4.4. Concept set similarity between sessions #10 and #5*

The set of concepts labeled for session #5 is;

$$URL_5 = \{\ 3,\ 7,\ 18,\ 34,\ 36,\ 52,\ 54,\ 57,\ 58,\ 199\ \}$$

The concept similarity matrix for those concepts labeled for sessions #10 and #5 is given in Figure 4.4. Hence the similarity between sessions #10 and #5 can be calculated as;

$$sim_{jaccard}(URL_{10}, URL_5) = \frac{7{,}18}{12} = 0{,}60$$

By comparing set similarity values, we can conclude that among its *k-nn* sessions, test session #10 is much more similar to session #5 with an accuracy rate %60.

Accuracy distribution for all test sessions is given in Figure 4.5; the average accuracy rate for test dataset is 66%.

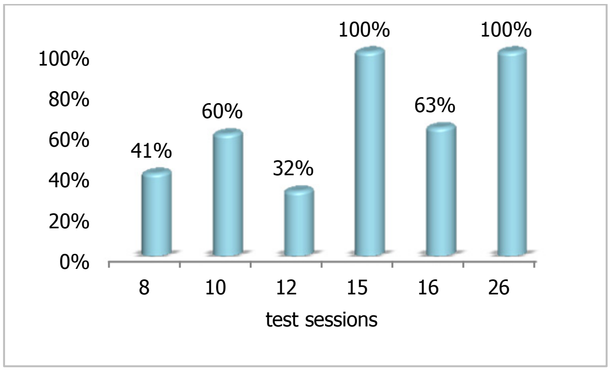
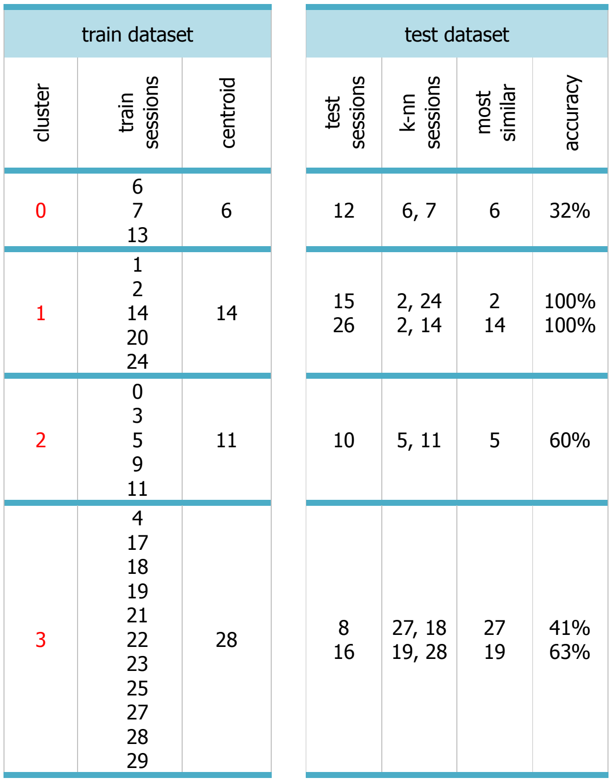


*Figure 4.5. Accuracy rates for test sessions*

Table 4.1 stands for the summary of all findings in our algorithm;

Table 4.1. *Findings for the sample dataset*



| train dataset | | | test dataset | | | |
|---|---|---|---|---|---|---|
| cluster | train sessions | centroid | test sessions | k-nn sessions | most similar | accuracy |
| 0 | 6<br>7<br>13 | 6 | 12 | 6, 7 | 6 | 32% |
| 1 | 1<br>2<br>14<br>20<br>24 | 14 | 15<br>26 | 2, 24<br>2, 14 | 2<br>14 | 100%<br>100% |
| 2 | 0<br>3<br>5<br>9<br>11 | 11 | 10 | 5, 11 | 5 | 60% |
| 3 | 4<br>17<br>18<br>19<br>21<br>22<br>23<br>25<br>27<br>28<br>29 | 28 | 8<br>16 | 27, 18<br>19, 28 | 27<br>19 | 41%<br>63% |

The algorithm detailed above is applied on a live web log dataset which is detailed in Section 4.1. The parameters used for clustering are 5-cluster, 5-knn and 5-fold. The overall accuracy rates are given in the graph in Figure 4.6. The average accuracy rate is %76.

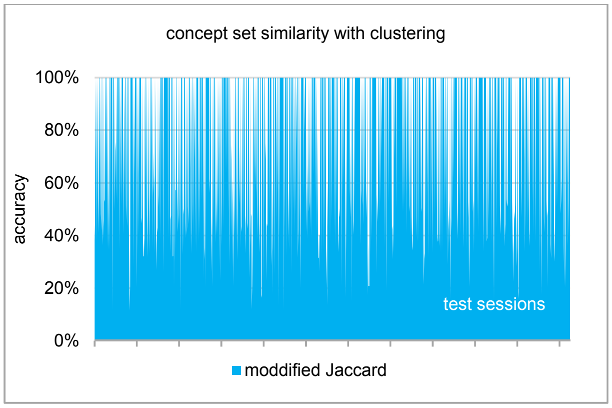


*Figure 4.6. Test results for accuracy rates on web log*[3]

## 4.3 Comparison With Other Approaches

For comparison, the following three options are tested against our "*cluster based with set similarity*" approach;

- URL-match with no-clustering
- URL-match with clustering
- concept set similarity with no-clustering

With all three, it is assumed that the dataset is already split into k-folds, train and test datasets are formed and *k-nn* sessions are located.

### 4.3.1 URL-match with no-clustering

The first alternative is to check the last URLs of *k-nn* sessions and see whether any of them confirms with the last URL of the test session that is being compared.

```
for test in tests:
    # get urls from k nearest neighbor
    nearestURLs = set()
    for s in knn:
        nearestURLs.add(sessions[s].pageviews[-1][0].url)

    testLastURL=sessions[test].pageviews[-1][0].url

    if testLastURL in nearestURLs:
        true_recom += 1
    else: false_recom += 1

return true_recom / (true_recom+false_recom)
```

*Figure 4.7.* *URL-match with no-clustering*

Simply it is done as follows:

- for each test session find the *k-nn* most similar sessions
- compare last URLs of these *k-nn* sessions with the last URL of the current test session
- if test session URL is one of those URLs then increase the count

For testing purposes, with 5-fold, 5-nn and 50 runs, we observed the hit rates shown in Figure 4.8. It should be noted that "one-to-one" comparison of each test session data against to all *k-nn* similar training dataset sessions is not practical in real implementations due to high processing cost. The average accuracy rate is 70%.

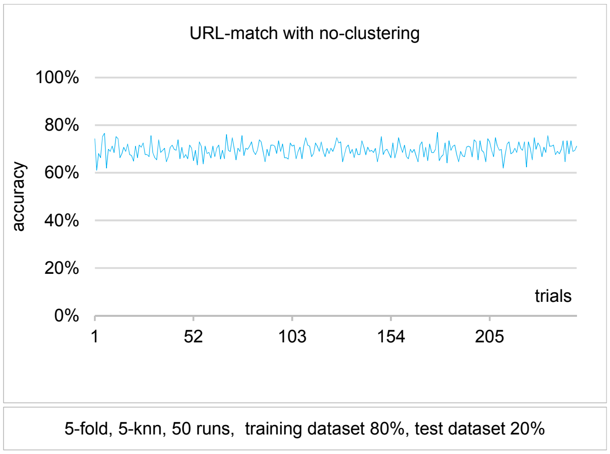
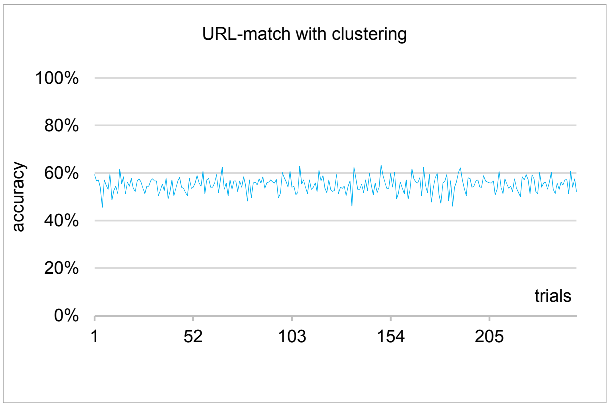


*Figure 4.8. URL-match with no-clustering*

### 4.3.2 URL-match with-clustering

Due to limitations in real life implementations – like reducing the cost of computation, for instance – next improvement is to create clusters among the training set together with determining their corresponding centroids. The motivation is to reduce search time and computation costs by just comparing each session with the pre-calculated cluster centroids rather than traversing all sessions in the training set. This approach significantly reduces the time for search as the search space would consist of only cluster centroids. For an initial test for cluster-centroid approach, the similarity comparisons are first accomplished by URL-matching between test session URLs and "ready-to-use" centroid sessions' URLs.

Based on our dataset, the average accuracy dropped to 55%. Compared to URL matching with no-clustering, there has been roughly 15% decrease in prediction accuracy. The obvious reason for the decrease is due to clustering which eliminates some more similar sessions out of the cluster. On the other hand, clustering brings low cost in terms of time efficiency.

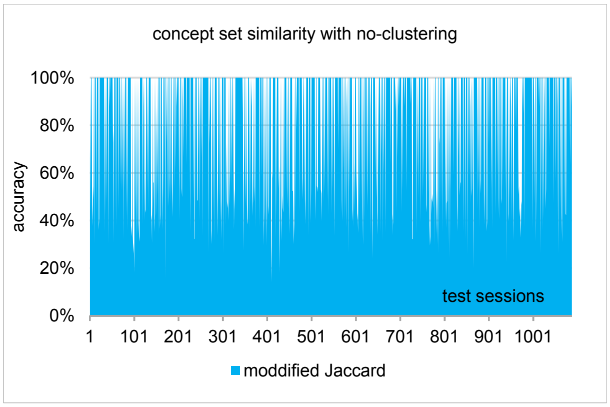


5-fold, 5-knn, 50 runs,  training dataset 80%, test dataset 20%

*Figure 4.9.* *URL-match with clustering*

### 4.3.3 Set similarity with no-clustering

The last option is similar to the previous one but this time test sessions are compared with their label sets, i.e. set of concepts, rather than relying upon only URL-strings. The idea behind is that the URL approach is not a standing attribute that best describes the content of the webpage.

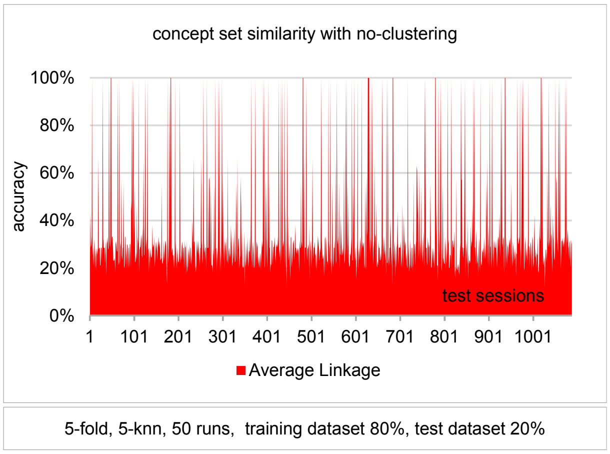


5-fold, 5-knn, 50 runs,  training dataset 80%, test dataset 20%

*Figure 4.10.* *Modified Jaccard concept set similarity with no-clustering*

Set of concepts associated with each webpage brings more value in terms of mining process for a better "*content-based*" prediction. The accuracy rate is raised up to 82%. Compared to "URL matching without clustering", there has been roughly 12% improvement in prediction accuracy.

As already mentioned in the thesis statement in the abstract section, the success of the prediction is highly correlated with the similarity function chosen. To illustrate this fact, an additional test has been carried out using the well-known similarity function "Average Linkage". Based on the same dataset, the average accuracy dropped as close as to 34% for Average Linkage similarity.



5-fold, 5-knn, 50 runs, training dataset 80%, test dataset 20%

*Figure 4.11. Average Linkage concept set similarity with no-clustering*

## 4.4 Time efficiency analysis

Algorithms for four different approaches mentioned above yield different time efficiencies. One of the claims in this thesis is that while the accuracy rates drop slightly when switching to clustering option for both URL-match and concept set similarity approaches, clustering is expected to compensate this accuracy loss with a better time efficiency. Tests on the dataset proved this fact.

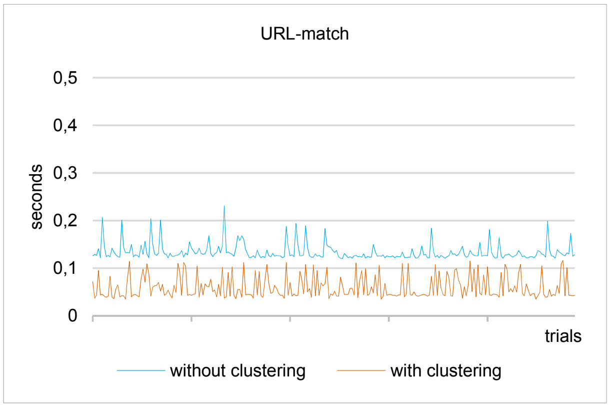


*Figure 4.12. Clustering effect on URL-match approach in terms of execution time*

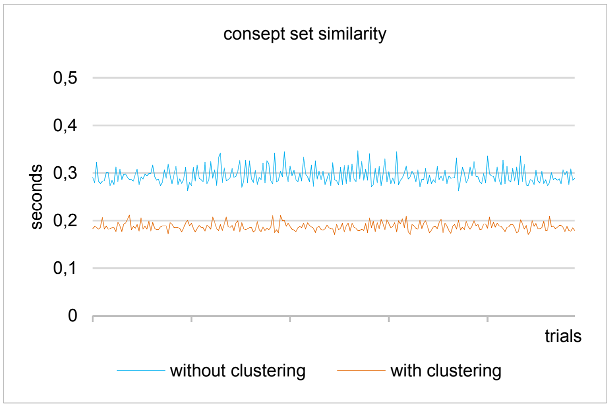


*Figure 4.13. Clustering effect on concept set similarity approach in terms of execution time*

Execution times are measured using all 1,126 sessions in the dataset with partitions %80 training vs %20 test respectively. Hence, time values shown in Figure 4.12 and Figure 4.13 are for the execution time for 226 test sessions to be predicted against 900 training sessions.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

In this study, we have investigated the problem of determining the next page that might be visited by the web user. Our proposed solution makes use of the contents of the pages rather than just using the URLs in order to be able to estimate the intent of the user by considering the semantics of the web pages. Furthermore, rather than searching all the previously recorded sessions to find sessions with similar navigation behaviour of the user, we propose to cluster previous sessions based on their similarities and for the current user navigation, the next possible page is predicted by first searching the most similar cluster and then by searching the most similar sessions in that cluster. This reduces the amount of the search time while dropping the accuracy slightly in an acceptable level. Table 5.1 shows the accuracies obtained by four different methods introduced in the study.

*Table 5.1. Comparison of prediction accuracy rates*

| | without clustering | with clustering |
|---|---|---|
| URL-match | 70 % | 55 % |
| concept set similarity | 82 % | 76 % |

Although the accuracy rates drop when clustering is applied, the gain in the execution times due to the clustering is worthwhile for this loss. More specifically, while the accuracy rates decrease with clustering option for both approach –URL-matching and concept set similarity–, in our experiments we have almost managed to double the speed of the prediction process. Table 5.2 summarizes execution times for the algorithms applied to web access log dataset.

*Table 5.2. Comparison of prediction execution times (seconds)*

| | without clustering | with clustering |
|---|---|---|
| URL-match | 0.134 | 0.059 |
| concept set similarity | 0.294 | 0.187 |

As a future work, the following issues should be investigated: it is not possible to keep all previously recorded sessions, so a mechanism should be devised to choose which sessions are going to be kept. Some simple alternatives are first come first leave, least frequent ones leave, and sessions older than some time frame leave, or some combinations of those strategies. Another issue is to consider the user information, when it is known (by using her IP). It might be possible do adjust the system using this information for a better prediction.

# REFERENCES

[1] Cyrus Shahabi, Farnoush Babaei-Kashani, Yi-Shin Chen, and Dennis McLeod "*Yoda: An Accurate and Scalable Web-based Recommendation System*", Proceeding of the Sixth International Conference on Cooperative Information Systems, Trento, Italy, September 2001

[2] Thwe, P. 2014. "*Web Page Access Prediction based on an Integrated Approach*", International Journal of Computer Science and Business Informatics, Vol. 12, No. 1, pp. 55-64.

[3] M. Erinaki, M. Vazirgiannis, I. Varlamis, "*Using site semantics and a taxonomy to enhance the web personalization process*"

[4] M. Perkowitz, O. Etzioni, "*Adaptive Web Sites: Automatically Synthesizing Web Pages"*, in Proc. of the 15th National Conference on Artificial Intelligence, Madison, WI, July 1998

[5] M. Perkowitz, O. Etzioni, "*Adaptive Web Sites: Conceptual Cluster Mining*", in Proc. of the 16th International Joint Conference on Articial Intelligence (IJCAI99), Stockholm, Sweden, 1999

[6] M. Perkowitz, O. Etzioni, "*Towards Adaptive Web Sites: Conceptual Framework and Case Study*", in Proc. of WWW8, 1999

[7] M. Eirinaki, M. Vazirgiannis, "*Web Mining for Web Personalization, ACM Transactions on Internet Technology (TOIT)*", February 2003/ Vol.3, No.1, 1-27

[8] B. Mobasher, R. Cooley, J. Srivastava, "*Creating Adaptive Web Sites Through Usage-Based Clustering of URLs*", in Proc. of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99), November 1999

[9] B. Mobasher, R. Cooley, J. Srivastava, "*Automatic Personalization Based on Web Usage Mining*", Communications of the ACM, August 2000/Vol. 43, No. 8, 142-151

[10] B. Mobasher, H. Dai, T. Luo, Y. Sung, J. Zhu, "*Discovery of Aggregate Usage Profiles for Web Personalization*", in Proc. of the Web Mining for E-Commerce Workshop (WEBKDD'00), Boston, MA, August 2000

[11] B. Mobasher, H. Dai, T. Luo, Y. Sung, J. Zhu, "*Integrating Web Usage and Content Mining for More Effective Personalization*", in Proc. of the International Conference on E-Commerce and Web Technologies (ECWeb2000), Greenwich, UK, September 2000

[12] B. Berednt, M. Spiliopoulou, "*Analysis of navigation behaviour in web sites integrating multiple information systems*", The VLDB Journal (2000) 9, 56-75

[13] Wen-Wei Chen, "*Data Warehouse and Data Mining Tutorial*", Beijing: Tsinghua University Press, 2008, 4

[14] WANG Yong-gui, JIA Zhen "*Research on Semantic Web Mining*", 2010 International Conference On Computer Design And Appliations (ICCDA 2010)

[15] L. Getoor, Link Mining: "*A New Data Mining Challenge*", SIGKDD Explorations, vol. 4, issue 2, 2003.

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

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

[18] Varelas Ioannis "*Semantic Similarity Methods in WordNet and Their Application to Information Retrieval on the Web*", June 30, 2005

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

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

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

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

[23] Vladimir Likic, Ph.D. "*The Needleman-Wunsch algorithm for sequence alignment",* 7th Melbourne Bioinformatics Course, Bio21 Molecular Science and Biotechnology Institute The University of Melbourne

# APPENDIX A

# ONTOLOGY

The concept taxonomy used in this study is given below with associated keywords. The root of the tree is the concept called "*Thing*" which is the top most parent virtual concept. The depth of each concept is equivalent to the indentation level in the taxonomy, for instance *Person* has a depth 1, *Worker* has depth 2 and *AssistantProfessor* has depth 4. Sets of words next to each concept represent keywords associated with that concept. For details, see Section 3.1.1.

```
Person
    Worker
     Faculty
          Professor (#Professor)
               AssistantProfessor (#Assistant*Professor)
                    akyuz (#Oguz*Akyuz) (#Ahmet #Oguz #Akyuz)
                    tcan (#Tolga*Can)
                    skalkan (#Sinan*Kalkan)
                    manguoglu (#Murat*Manguoglu)
                    erol (#Erol*Sahin)
                    karagoz (#Pinar*Karagoz) (#Pinar*Senkul)
               AssociateProfessor (#Associate*Professor) (#Assoc.*Prof)
                    alpaslan (#Ferda #Alpaslan)
                    bozsahin (#Cem*Bozsahin)
                    cosar (#Ahmet*Cosar)
                    nihan (#Nihan*Cicekli)
                    dogru (#Ali*Dogru)
                    isler (#Veysi*Isler)
                    oguztuzn (#Halit*Oguztuzun)
               FullProfessor (#Full #Professor)
```

```
            volkan (#Volkan*Atalay)
            bozyigit (#Muslim*Bozyigit)
            asuman (#Asuman*Dogac)
            genc (#Payidar*Genc)
            ayse (#Ayse*Kiper)
            polat (#Faruk*Polat)
            sibel (#Sibel*Tari)
            toroslu (#Ismail*Toroslu)
            ucoluk (#Gokturk*Ucoluk)
            vural (#Fatos*Vural)
            yazici (#Adnan*Yazici)
            yalabik (#Nese*Yalabik)
    Lecturer (#Lecturer)
    PostDoc
    Doctor (#phd) (#doctorate)
        birturk (#Aysenur*Birturk)
        ruken (#Ruken #Cakici)
        ozgit (#Attila #Ozgit)
        onur (#Onur #Tolga #Sehitoglu)
        sener (#Cevat #Sener)
        faruk (#Faruk #Tokdemir)
Assistant (#assistant)
    ResearchAssistant (#research #assistant)
    TeachingAssistant (#teaching #assistant)
        okan (#Okan #Akalin)
        rusen (#Rusen #Aktas)
        merve (#Merve #Aydinlar)
        levent (#Levent #Bayindir)
        hande (#Hande #Celikkanat)
        sciftci (#Serdar #Ciftci)
        sinem (#Sinem #Demirci)
        deniz (#Onur #Deniz)
        odulger (#Ozcan #Dulger)
        eksert (#Levent #Eksert)
        alperen
        asli
        genctav
        fgokce (#Fatih #Gokce)
        gulen (#Elvan #Gulen)
        kerem (#Kerem #Hadimli)
```

hosgor (#Can #Hosgor)
gokdeniz (#Gokdeniz #Karadag)
mckaya
okaya (#Ozgur #Kaya)
ketenci (#Ahmet #Ketenci)
akilic (#Alper #Kilic)
hkilic (#Hilal #Kilic)
sefa (#Sefa #Kilic)
celebi (#Celebi #Kocair)
mutlu
burcak
anil (#Anil #Sinaci)
erdal (#Erdal #Sivri)
selma (#Selma #Suloglu)
ftitrek (#Fatih #Titrek)
tarhan (#Okan #Tarhan)
gtumuklu (#Gulsah #Tumuklu)
aysegul
hyildiz (#Husnu #Yildiz)
myoldas (#Mine #Yoldas)
cuneyt
marlen
ahmet
alan (#Ozgur*Alan)
bugra (#Bugra*Ozkan)
Student (#student)
UndergraduateStudent (#undergraduate #student)
GraduateStudent (#graduate #student)
Publication (#publication)
Article (#article)
JournalArticle (#journal #article)
ConferencePaper (#conference #paper)
Book (#book)
Manual (#manual)
Periodical
Journal (#journal)
Magazine (#magazine)
Proceedings (#proceeding)
Specification (#specification)
TechnicalReport (#technical #report)

```
    Thesis (#thesis)
     DoctoralThesis (#doctoral #thesis)
     MastersThesis (#master #thesis)
    UnofficialPublication (#unofficial #publication)
Work
    Course (#course)
     MustCourse (#must #course)
           ceng100 (#ceng*100)
           ceng111 (#ceng*111)
           ceng140 (#ceng*140)
           ceng213 (#ceng*213)
           ceng223 (#ceng*223)
           ceng232 (#ceng*232)
           ceng242 (#ceng*242)
           ceng280 (#ceng*280)
           ceng300 (#ceng*300)
           ceng315 (#ceng*315)
           ceng331 (#ceng*331)
           ceng334 (#ceng*334)
           ceng336 (#ceng*336)
           ceng350 (#ceng*350)
           ceng351 (#ceng*351)
           ceng378 (#ceng*378)
           ceng382 (#ceng*382)
           ceng400 (#ceng*400)
           ceng436 (#ceng*436)
           ceng477 (#ceng*477)
           ceng491 (#ceng*491)
           ceng492 (#ceng*492)
     TechnicalElectiveCourse (#technical #elective)
           ceng210 (#ceng*210)
           ceng220 (#ceng*220)
           ceng316 (#ceng*316)
           ceng332 (#ceng*332)
           ceng335 (#ceng*335)
           ceng340 (#ceng*340)
           ceng352 (#ceng*352)
           ceng356 (#ceng*356)
           ceng371 (#ceng*371)
           ceng372 (#ceng*372)
```

  - ceng373 (#ceng*373)
  - ceng424 (#ceng*424)
  - ceng437 (#ceng*437)
  - ceng443 (#ceng*443)
  - ceng444 (#ceng*444)
  - ceng451 (#ceng*451)
  - ceng452 (#ceng*452)
  - ceng462 (#ceng*462)
  - ceng463 (#ceng*463)
  - ceng465 (#ceng*465)
  - ceng466 (#ceng*466)
  - ceng469 (#ceng*469)
  - ceng476 (#ceng*476)
  - ceng478 (#ceng*478)
  - ceng483 (#ceng*483)
  - ceng493 (#ceng*493)
  - ceng497 (#ceng*497)
  - ceng498 (#ceng*498)
- ServiceCourse (#service #course)
  - ceng200 (#ceng*200)
  - ceng221 (#ceng*221)
  - ceng230 (#ceng*230)
  - ceng301 (#ceng*301)
  - ceng302 (#ceng*302)
  - ceng303 (#ceng*303)
  - ceng470 (#ceng*470)
  - ceng494 (#ceng*494)
- GraduateCourse (#graduate #course)
  - ceng500 (#ceng*500)
  - ceng508 (#ceng*508)
  - ceng520 (#ceng*520)
  - ceng530 (#ceng*530)
  - ceng531 (#ceng*531)
  - ceng532 (#ceng*532)
  - ceng534 (#ceng*534)
  - ceng535 (#ceng*535)
  - ceng536 (#ceng*536)
  - ceng538 (#ceng*538)
  - ceng540 (#ceng*540)
  - ceng545 (#ceng*545)

ceng546 (#ceng*546)
ceng550 (#ceng*550)
ceng551 (#ceng*551)
ceng553 (#ceng*553)
ceng554 (#ceng*554)
ceng555 (#ceng*555)
ceng556 (#ceng*556)
ceng557 (#ceng*557)
ceng558 (#ceng*558)
ceng559 (#ceng*559)
ceng561 (#ceng*561)
ceng562 (#ceng*562)
ceng563 (#ceng*563)
ceng564 (#ceng*564)
ceng565 (#ceng*565)
ceng566 (#ceng*566)
ceng567 (#ceng*567)
ceng568 (#ceng*568)
ceng569 (#ceng*569)
ceng571 (#ceng*571)
ceng572 (#ceng*572)
ceng573 (#ceng*573)
ceng574 (#ceng*574)
ceng575 (#ceng*575)
ceng576 (#ceng*576)
ceng577 (#ceng*577)
ceng580 (#ceng*580)
ceng581 (#ceng*581)
ceng582 (#ceng*582)
ceng583 (#ceng*583)
ceng584 (#ceng*584)
ceng585 (#ceng*585)
ceng591
ceng701
ceng708
ceng710
ceng712
ceng713
ceng714
ceng729

  - ceng732
  - ceng734
  - ceng740
  - ceng763
  - ceng764
  - ceng769
  - ceng770
  - ceng771
  - ceng784
- MSCENGwoThesisCourse
  - ceng508_2
  - ceng525
  - ceng532_2
  - ceng536_2
  - ceng538_2
  - ceng546_2
  - ceng553_2
  - ceng561_2
  - ceng562_2
  - ceng564_2
  - ceng567_2
  - ceng577_2
  - ceng599
  - ceng707
  - ceng709
  - ceng714_2
- MSSEwoThesisCourse
  - se448
  - se541
  - se542
  - se546
  - se547
  - se548
  - se550
  - se554
  - se560
  - se599
  - se704
  - se705
  - se706

Research (#research)

ResearchLaboratory (#research #lab) (#research #laboratory)

BioinformaticsLab (#bioinformatics #lab) (#bioinformatics #laboratory)

MultimediaDatabase (#multimedia #database #research) (#multimedia #database #laboratory) (#multimedia #database #lab)

ImageProcessing (#image #processing #lab) (#image #processing #laboratory) (#pattern #recognition #lab) (#pattern #recognition #laboratory)

ISL (#intelligent #system #lab) (#intelligent #system #laboratory)

Kovan (#robotics #lab) (#robot #lab) (#robotics #laboratory) (#robot #laboratory)

LCSL (#computational #studies #language) (#computational #study #language)

ParallelProcessing (#parallel*processing)

ResearchGroup (#research*group)

ComputerGraphics (#graphics) (#visualization)

DataMining (#data*mining)

EvolutionaryComputing (#evolutionary)

GridComputing (#grid*computing) (#grid #compute)

ResearchAssociatedCenter (#research*center)

Modsim (#modeling #simulation)

SRDC

Schedule (#schedule)

Resources (#resource)

ComputingServices (#computing*service)

Documents

StudentDocuments (#student #doc) (#student #documents)

StaffDocuments (#private #staff #doc) (#private #document) (#staff #document)

NewsArchive (#news #archive) (#anouncement)

Seminar (#seminar)

CSTopic

DataStructures (#stack #queue) (#tree*structure) (#hash) (#data*structure) (#abstract*data)

Algorithms (#sorting #algorithm) (#search #algorithm) (#graph #algorithm) (#algorithm #complexity) (#np #completeness) (#divide #conquer) (#dynamic #programming)

DiscreteMath (#proposition) (#predicate #logic) (#set #theory) (#induction) (#discrete #math) (#discrete #mathematics)

TheoryComp (#theory #computation) (#automata) (#pushdown) (#deterministic #automata) (#nondeterministic #automata) (#regular*expression) (#Turing*machine) (#Church #Turing) (#halting*problem) (#pumping*lemma) (#context #free #grammar) (#contextfree #grammar)

PL (#programming*language) (#functional*language) (#object #oriented) (#imperative #programming) (#python) (#java) (#haskell) (#encapsulation) (#inheritance) (#operator #overload) (#parameter #passing)

OS (#operating*system) (#process #thread) (#deadlock) (#synchronization) (#interprocess*communication) (#memory*management) (#file*system) (#semaphore) (#virtual*memory)

Digital (#circuit #digital) (#register #memory) (#arithmetic #logic #unit) (#interrupt #hardware)

AI (#artificial*intelligence) (#heuristic #algorithm) (#concept #learning) (#reasoning) (#reinforcement #learning) (#supervised #learning) (#game*playing) (#logical*inference) (#knowledge #representation)

Graphics (#computer #graphics) (#geometry #transformation) (#render #graphics) (#projection #graphics) (#clipping #graphics) (#light #graphics) (#reflection #graphics) (#texture #graphics) (#ray #graphics) (#surface #graphics) (#shading) (#polygon) (#opengl) (#glut) (#glui)

NLP (#natural*language) (#natural #language #processing) (#morphology) (#linguistic) (#parsing #language)

Database (#database #management #system) (#relational*algebra) (#sql) (#transaction #database) (#database #integrity) (#database #query) (#data*model) (#er #diagram) (#entity #relation) (#dbms)

SoftwareEng (#software #engineering) (#project #management) (#software #quality) (#software #integration) (#software #maintenance) (#process #model) (#software #testing) (#quality #assurance)

PatternRecognition (#pattern #recognition) (#pattern #classification) (#bayes) (#neural #networks) (#clustering) (#decision #theory) (#feature #selection)

ParellelComputation (#parallel #computing) (#parallel #computation) (#parallel #performance) (#parallel #algorithms)

Bioinformatics (#bioinformatics) (#microarray) (#sequence*alignment) (#gene #network) (#protein #network) (#protein #structure) (#phylogenetic #tree)

# APPENDIX B

# SESSION SIMILARITY MATRIX

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | .09 | .06 | .18 | .14 | .40 | .05 | .07 | .15 | .24 | .31 | .22 | .09 | .08 | .08 | .07 | .13 | .14 | .18 | .20 | .07 | .17 | .12 | .14 | .10 | .14 | .05 | .17 | .14 | .12 |
| 1 | | .68 | .54 | .57 | .21 | .27 | .25 | .39 | .23 | .25 | .29 | .51 | .25 | .60 | .68 | .42 | .50 | .46 | .43 | .39 | .39 | .41 | .36 | .60 | .41 | .62 | .53 | .60 | .50 |
| 2 | | | .43 | .40 | .16 | .24 | .25 | .21 | .21 | .22 | .25 | .50 | .20 | .69 | 1,00 | .23 | .44 | .35 | .33 | .44 | .26 | .43 | .18 | .80 | .45 | .75 | .34 | .35 | .32 |
| 3 | | | | .41 | .37 | .14 | .19 | .45 | .39 | .41 | .59 | .33 | .21 | .22 | .21 | .37 | .37 | .53 | .39 | .23 | .42 | .34 | .42 | .27 | .41 | .16 | .51 | .45 | .41 |
| 4 | | | | | .34 | .19 | .28 | .52 | .38 | .29 | .41 | .36 | .21 | .23 | .32 | .60 | .57 | .55 | .67 | .26 | .60 | .53 | .45 | .29 | .61 | .21 | .55 | .66 | .75 |
| 5 | | | | | | .08 | .12 | .24 | .55 | .52 | .48 | .18 | .12 | .13 | .13 | .33 | .29 | .38 | .50 | .11 | .36 | .27 | .33 | .18 | .29 | .09 | .34 | .28 | .21 |
| 6 | | | | | | | .80 | .33 | .18 | .11 | .21 | .75 | .80 | .40 | .50 | .22 | .57 | .46 | .22 | .33 | .20 | .25 | .20 | .40 | .38 | .33 | .60 | .50 | .30 |
| 7 | | | | | | | | .36 | .21 | .16 | .24 | .65 | .51 | .34 | .39 | .28 | .66 | .53 | .29 | .34 | .32 | .37 | .26 | .38 | .46 | .25 | .56 | .53 | .39 |
| 8 | | | | | | | | | .38 | .28 | .51 | .27 | .34 | .26 | .23 | .46 | .56 | .76 | .43 | .26 | .57 | .42 | .61 | .30 | .44 | .18 | .83 | .67 | .64 |
| 9 | | | | | | | | | | .48 | .59 | .23 | .19 | .17 | .15 | .46 | .47 | .52 | .47 | .16 | .50 | .35 | .49 | .22 | .42 | .13 | .47 | .46 | .36 |
| 10 | | | | | | | | | | | .49 | .19 | .11 | .12 | .12 | .37 | .38 | .44 | .40 | .12 | .36 | .24 | .42 | .19 | .32 | .09 | .40 | .36 | .27 |
| 11 | | | | | | | | | | | | .23 | .18 | .18 | .18 | .51 | .44 | .58 | .46 | .20 | .45 | .31 | .45 | .23 | .41 | .13 | .61 | .51 | .41 |
| 12 | | | | | | | | | | | | | .20 | .18 | .23 | .24 | .38 | .30 | .28 | .26 | .25 | .31 | .20 | .20 | .50 | .20 | .26 | .28 | .25 |
| 13 | | | | | | | | | | | | | | .47 | .40 | .20 | .54 | .54 | .20 | .33 | .32 | .28 | .28 | .47 | .43 | .27 | .67 | .47 | .36 |
| 14 | | | | | | | | | | | | | | | .80 | .18 | .46 | .50 | .22 | .41 | .28 | .37 | .23 | 1,00 | .37 | .80 | .50 | .32 | .28 |
| 15 | | | | | | | | | | | | | | | | .23 | .45 | .36 | .35 | .49 | .28 | .43 | .18 | .80 | .45 | .62 | .36 | .38 | .34 |
| 16 | | | | | | | | | | | | | | | | | .73 | .66 | .89 | .22 | .52 | .40 | .58 | .19 | .66 | .16 | .59 | .74 | .52 |
| 17 | | | | | | | | | | | | | | | | | | .80 | .52 | .29 | .48 | .41 | .40 | .23 | .56 | .21 | .60 | .75 | .52 |
| 18 | | | | | | | | | | | | | | | | | | | .54 | .23 | .53 | .36 | .58 | .26 | .51 | .17 | .80 | .62 | .45 |
| 19 | | | | | | | | | | | | | | | | | | | | .22 | .52 | .54 | .43 | .24 | .58 | .19 | .44 | .55 | .42 |
| 20 | | | | | | | | | | | | | | | | | | | | | .24 | .28 | .19 | .29 | .36 | .29 | .28 | .32 | .26 |
| 21 | | | | | | | | | | | | | | | | | | | | | | .44 | .47 | .29 | .53 | .15 | .53 | .55 | .67 |
| 22 | | | | | | | | | | | | | | | | | | | | | | | .32 | .31 | .47 | .21 | .36 | .41 | .42 |
| 23 | | | | | | | | | | | | | | | | | | | | | | | | .23 | .55 | .14 | .85 | .71 | .47 |
| 24 | | | | | | | | | | | | | | | | | | | | | | | | | .36 | .54 | .42 | .31 | .29 |
| 25 | | | | | | | | | | | | | | | | | | | | | | | | | | .20 | .42 | .53 | .44 |
| 26 | | | | | | | | | | | | | | | | | | | | | | | | | | | .45 | .38 | .30 |
| 27 | | | | | | | | | | | | | | | | | | | | | | | | | | | | .70 | .48 |
| 28 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | .63 |

*Figure B.1.* *Session similarity matrix*