Improving the user modeling in website using the semantic web and domain specific concepts

Mehdi Sadeghzadeh, Ph.D. (Corresponding author)1, Navid Alizadeh Kordestani2

1Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
sadegh1999@yahoo.com
2Msc Student of Computer Engineering, Department of Computer Engineering, Damavand Branch, Islamic Azad University, Damavand, Iran
websland@gmail.com

ABSTRACT
The information burdensome is one of the main problems in the current web environment. To repel such a problem, some web personalization systems have been provided which adjust the content and services of a website with the individuals based on their interests and web-mining behavior. The principal achievement of the present research is to introduce a mechanism for demonstrating the user in the website automatically and by means of a comprehensive lexical semantic source. Wikipedia, the biggest contemporary encyclopedia, has been exploited as a rich semantic source for improving the auto-modeling of the user’s interests. The provided architecture consists of some components including: primary preprocessing, extracting the website domain concepts, extracting the keywords from the website, keywords vector constructor, mapping the keywords into the concepts. Another important achievement is a new method for mapping the keywords into the concepts. Our evaluations indicate that the suggested method, together with the comprehensive lexical semantic source, represents the users more effectively compared to the keywords method and other WordNet-based methods.

Keywords: user modeling, Wikipedia mining, semantic techniques, web personalization

INTRODUCTION
Reasons for web personalization requirement
The global web has provided a vast source of information. Various studies related to the web development issue have estimated that the number of web pages increases as much as one million pages per day; besides, more than 600 gigabytes of the pages change every month (Nasraoui, 2008) & (Achananuparp, 2007). This phenomenon is called information burdensome which creates some problems for the web users.
One of the most important problems is lack of a convenient and easy access to the required information. In such a vast storage the users have difficulties in finding their needed information in a proper time and in an easy way because, on one hand, they should investigate the relevance of each page to their need and, on the other hand, they should evaluate the pages in terms of reliability. In recent decades in order to resolve this problem, the information (data) recovery systems and, consequently, the search engines have been created which index the content of web pages and recover the pages related to the user’s question.
Appearance of the web-based services such as e-commerce, web-based learning, and e-banking, has caused some fundamental changes in the web using method and has converted the websites into an environment for trading and commerce, and thus has increased the competition between these web pages. The existence of the rivals whose distance from the intended website is as long as only one click necessitates the improvement of the web services as a requirement of creating steadfast and permanent customers. Providing these extra services is possible only through concentration on the individual needs and interests of the users and providing relevant products and services for them.

Main requirements of web personalization system
The functions of the web personalization system demonstrate the requirements of designing such a system. This subject is explained in detail in chapter-2. These requirements include:
• Domain specifications (features)
The function presented by the personalization system is domain-sensitive thus the specifications of the domain must be described precisely.
• User recognition
The personalization system directly interacts with the user and gets information about his/her behavior
thus it should have a mechanism appropriate for recognizing and distinguishing the users.

- Efficient reception of the users’ information (data)
  The personalization system should have the capability to collect all the data related to the users. The volume and type of these data depends on the system functions.
  - Data preparation
    The collected data should be preprocessed in order to eliminate their noise and to convert them to a proper format.
  - Efficient creation of the user model
    The main component of the personalization system is the user model which includes the information about the user’s interests, knowledge, goals, and preferences. The model can be made automatically or manually.
  - Topics related to privacy
    The user’s information must be always preserved and secured and the user should be aware of the manner of collecting and using such information.

**Web personalization approaches**

The systems of web personalization can be categorized in three approaches (Dai, 2005). In this section, we will describe these categories summarily.

- Systems based on the law of manual decision
  Based on this approach a web service is personalized through its designer’s manual interference and, usually, its user’s cooperation. The static user models are usually obtained through an enrolling process and some laws (rules) are manually determined for the manner of delivering the web content to the users who have different models. An example of these systems is *WebSphere Personalization* provided by IBM Company.

- Content-based filtering systems
  These systems use the users’ profiles and recommend new items or pages to the user based on their similarity to the pages or items which exist in the user’s profile. The common mechanism in these systems is comparison of the keywords demonstrating the pages or description of the items. *Leticia* and *WebWatcher* are two examples of this kind of systems.

- Collective or social filtering systems
  The principal concentration of these systems is mainly on the similarity of the users rather than the similarity of the items. These systems compare the intended user’s preferences history with all the other users in order to find the users who have interests similar to the intended user’s interests. Such set of the users with similar interests is called *current user neighborhood*. Mapping between a user’s history and his neighbors can be done based on the similarity of the items ranking, access to the pages with similar content, or purchasing similar items. Then the obtained neighborhood is used to recommend the items which have been accessed or purchased by the current user.

**Role of web usage mining in web personalization**

Generally web mining can be considered as data mining on the data related to the web content, structure, and usage. The aim of web mining is to discover the models and patterns lied in the web sources. The web usage mining is specifically aimed to discover the behavioral patterns of the web users. Discovering such patterns from among a huge volume of data produced by the web servers has got important usages and applications (Anand, 2005); for example, systems which evaluate a site’s effectiveness level in fulfilling the user’s expectations; techniques for dynamically balancing the loads and optimizing the web servers to provide the users with more effective access; and some other applications related to reconstructing and adapting a site based on the user’s predicted needs.

**Need for using the content in web personalization**

An approach which is based merely on the usage has an important deficiency; that is, the process of recommending an item to the user is done based only on his existing transactional data and thus the pages or items which have been recently added to the site can’t be recommended to the user. This problem is called the *new item problem*. On the other hand, although the discovered patterns related to using the web sources through web usage mining are helpful in discovering the relationships of the items with each other or the relationships of the users with each other and also in determining the similarity of the users’ sessions, but these patterns, without a deep knowledge of the intended website domain, can rarely help us to understand why some items or users are put together in a group or category. A common approach to solve this problem in collective filtering is to merge the features (characteristics) of the pages’ content with the rankings and judgments of the users.

**Need for using the semantics in web personalization**

The keyword-based approaches are feeble in perceiving the complicated relationships between the objects in deeper semantic levels. For example, the valuable information of the films, directors, and actors or the students, courses, and professors will be missed if only the keywords are used to describe these entities. In order to recommend different kinds of the complex objects using their features and characteristics, the system should be able to consider their features in a semantic depth higher than that of the keywords. The traditional web personalization system of a university may recommend the JAVA courses to the student just because a student has previously shown his interest in these courses. Furthermore, a system which exploits the *relevant domain knowledge* may recognize that this student must pass the JAVA
course prerequisites or it may have the capability to recommend to the student the most appropriate professor for this course. Ontology is a philosophical discussion which defines a structure for defining the important concepts and the semantic relationships between them. This structure, when taken down in machine, is called lexicology. An example of lexicology is a relational schema in database including tables and outer keys which are semantically interrelated. Such structures can be used to produce knowledge in high abstract levels in a specific domain. Lexicology for a website usually embraces concepts, ranking relationships between concepts, and other relationships between the concepts in a website's domain. For example, the lexicology of the domain of a film website usually includes concepts such as movie, actor, director, and so on. The usual relationships in this domain include playing in a movie (between movie & actor), directing, and so on.

Need for using semantics in user demonstration
One of the principal requirements of a personalization system is efficient creation of the user model. This component helps the system to achieve a good perception of various aspects of its users such as their background, interests, and so on. The user models can be categorized in different ways and based on different aspects (Achananuparp, 2007), e.g. short-term or long-term, explicit or implicit, group or individual, and so on. In chapter 3 we will discuss in detail about the user modeling in website personalization.

The main weak point in most of the approaches which use the web content for improving the user model is that these approaches usually use the expressions vector to demonstrate the user’s interests and ignore the semantic relationships between these expressions while using semantics can improve the demonstration method. The main goal of this project is to improve the user model using the semantics of the website domain in websites which use the web content, so it is desirable that this process is done automatically as much as possible.

Suggested method
In this section we present the suggested method for improving the vector space model of the user’s behavior in website. First a precise definition of the problem is presented and a checklist of the desired features of the model is provided. Then the system architecture is presented and its components and performance are explained.

The goal is to provide a method for improving the vector space model of keywords based on the user’s behavior in the website, a method which has the following features:
- The model should be created implicitly, that is without the user’s direct interference.
- The model should be individual; that is, there should be a specific model for each user.
- The model should be created based on the user’s mining behavior in a specific time period, for example during his two-week mining in the website.
- The process of creating the model should be automatic as much as possible.
- The model should contain domain specific concepts together with their values for the user’s visits.

More formally, suppose that the user $U$ has visited the website pages in $T$ time period and has had $\{s_1, s_2, \ldots, s_m\}$ sessions. The goal is to make a vector of the domain’s important concepts and their weights, that is, $V = \langle(c_1, w_1), (c_2, w_2), \ldots, (c_n, w_n)\rangle$ such that the concepts are chosen appropriately (i.e. they have good precision and recall) and their weights demonstrate their importance in the user’s behavior during the given time period as much as possible.

Designing a new method
The architecture of the suggested system is shown in figure-1. As it can be seen in figure-1, the system includes some components which will be explained in detail in the following sections.
Figure 1: the general architecture of the suggested system

Log primary preprocess component
The duty of this component is to apply the primary preprocesses of the web server’s logs such as data cleaning, user recognition, user’s session recognition, and so on. The input of this component includes the web server’s logs, and website pages and its output includes the users’ sessions.

Component of extracting the keywords from the website pages
The keywords in a website page are those words and expressions which are considered important for that page. These keywords usually include special words expressions, nominated entities, new expressions, and other words related to the content. Extracting the keywords is, in fact, the act of automatically recognizing the keywords in a page. There are various methods for extracting the keywords, including methods with supervisor and without supervisor. The methods with supervisor often apply the machine learning models such as Naïve Bayes, Decision Tree, etc., using features like syntactic features. The methods without supervisor, too, have demonstrated a good performance in this action. The method used in the present thesis is based on a method (Mihalcea, 2007) which is a learning method without supervisor using Wikipedia and can be considered as a state-of-the-art method for extracting the keywords. Since the criteria for choosing a link in Wikipedia is very similar to the criteria for choosing the keywords in documents, recognizing the keywords can be considered as linking the words in a Wikipedia article.
The candidate keywords should be limited to those ones which have a creditable corresponding article in Wikipedia. Thus we can construct a glossary for the keywords which includes only the Wikipedia articles topics related to the intended website domain, and can use this glossary for extracting the keywords; however, this restricts the capability of finding the keywords because the link labels are not necessarily the same as the articles’ topics. If these states are ignored we may miss a part of these words, thus we develop the primary glossary with all of the links’ labels. Details of the keywords extraction component which is without supervisor...
are shown in figure 2. This component is without supervisor in order to achieve a higher efficiency. It is consisted of two parts including candidate extraction and candidate ranking. The candidate extraction component processes the web pages and extracts all the n-grams in the glossary. The candidate ranking component allocates a numeric value to each candidate; this can indicate the probability of the candidate’s keyphraseness.

![Diagram showing the keywords extraction component](image)

**Figure 2: details of the keywords extraction component**

Using a limited glossary for extracting the keywords has the advantage that some of the problems and difficulties of other methods are avoided. For example, all the keywords in this glossary are acceptable expressions and there is no meaningless expression, e.g. “products are”. The pseudo-code of the candidate extraction section is shown in figure 3.

```
Function candidateExtraction()
    For each article a in filteredArticles
        For each link l in a
            If (l.article.title is in domainVocabulary
                and
                l.label is not in domainVocabulary)
                add l.label to domainVocabulary
            End if
        End for
    End for
End
```

**Figure 3: pseudo-code of the candidate words extraction component**

This probability can be interpreted in this way: the more a word in Wikipedia in its occurrences is selected as a link, the more the probability that it is again selected as a link. This probability, in fact, determines the weight of the keywords. The readers can refer to “Mihalcea, 2007” for observing this method. It must be noted that the glossary should be constructed in such a way that the topics and links of the articles are limited only to the intended domain. In the next section, we will explain that how we can automatically extract the domain articles from Wikipedia. The pseudo-code of the candidate ranking section is shown in figure 4.

\[
P(\text{keyword} \mid W) = \frac{\text{count}(W, \text{key})}{\text{count}(W)}
\]
The duty of this component is extracting the domain specific concepts from the website pages. In the present thesis, each Wikipedia article is considered as a concept, because each article concentrates on a specific event or entity and presents its related topics and important information with an appropriate structure. As shown in figure 5, this component, too, is consisted of two parts including candidate concepts and candidate filtering.

The candidate concepts extraction section is fundamentally based on the Wikipedia classification structure. Suppose that the intended website generally includes the IT domain. In the highest level of the classification structure there are 10 general classes among which the class of the website can be easily recognized. Then, automatically and by first-level searching, all the articles of that class can be obtained without performing the web-crawling action.

The concepts obtained from the candidate extraction section are numerous and, if not filtered, can decrease the system efficiency. Thus the candidate filtering section filters those concepts which are more relevant to the website using the website’s structure.

Most of the web documents in the websites are classified in a virtual hierarchical structure which classifies implicitly their content type (Chang, 2005). The directory structures indicate that the domain specific expressions in the texts of a particular subdirectory’s pages are closely dependent on a common topic which is specified in the directory’s name. This feature can be used to eliminate the concepts which are less relevant to the content of the website’s pages.

The candidate filtering section, first, extracts the important words (words which are not stopword) in the URL of the website’s pages and, then,
calculates the semantic relationship of each one through Extended Gloss Overlap method using the Wikipedia. The reason for using the text overlap approach is the simplicity of calculation due to lack of need for the hierarchical structure and, consequently, more efficiency. Every candidate concept whose semantic relationship with all the important words in the URL is equal to zero will be eliminated. The pseudo-code of the candidate filtering component is shown in figure 6.

As for calculating the \( \text{relatedness}_{\text{gloss}}(c,k) \), it must be noted that if a search for the word \( k \) in Wikipedia leads to an ambiguity removal page, then the \( \text{relatedness}_{\text{gloss}}(c,k) \) will show the highest value of the semantic relationship among all the ambiguity removal links.

![Figure 6: pseudo-code of the candidate filtering section](image)

**Keywords vector constructor component**
The duty of this component is to extract the keywords together with their appropriate weights from the user’s sessions. The output of the keywords extraction component for a page such as \( p_i \) is as follows:

\[
(2) \quad p_i = \langle (k_{i1}, w_{i1}), ..., (k_{in}, w_{in}) \rangle
\]

where the weights of the words are the same as their keyphraseness. Suppose that the user has had the following session:

\[
(3) \quad s = \{(p_1, t_1), ..., (p_s, t_s)\}
\]

where \( t_i \) is the time spent by the user in the page \( p_i \). We define the keywords vector for the user’s session as follows:

\[
(4) \quad s = \langle (k_{1}, w_{1}), ..., (k_{s}, w_{s}) \rangle
\]

where

\[
(5) \quad k_i = \langle k_{i1}, ..., k_{ij} \rangle
\]

\[
(6) \quad w_i = \langle w_{i1}, ..., w_{ij} \rangle
\]

\[
(7) \quad w_{ij} = \text{keyphraseness}(k_i) \times \frac{t_{ij}}{s_{\text{max}}}
\]

\( s_{\text{max}} \) is the maximum time length of the session which is supposed 30 minutes. The relationship (7) indicates that the more time the user spends on a page the more importance the keywords of that page find for him. Note that the time spent on the last page of the session can’t be obtained from the web server log and should be estimated. The weight of the keyword in the whole session is equal to its total weight in all the keywords vectors:

\[
(8) \quad \text{weight}(s,k) = \sum_{k_{ij} = k} w_{ij}
\]

**Mapping component**
The duty of this component is to construct the user model through mapping the keywords of the pages visited by the user in the given time period into the domain filtered concepts. This step is aimed to find the most relevant concepts for the keywords. This concept may be the precise concept correspondent with the word or a more abstract concept. To achieve this goal, we use the path-based semantic relationship approach since this approach considers the implicit semantic relationship in the hierarchical structure such as hypernymy and hyponymy and, in case of not finding the precise concept correspondent with the keyword, finds its superclass.

In order to perform mapping, we calculate the semantic relationship of the keywords of the pages in the user’s sessions individually with the filtered concepts and then attribute the concept having the most semantic relationship to each keyword. Finally, we obtain the weight of each concept from the total weight of all the keywords to which this concept has been attributed. The pseudo-code of this algorithm is shown in figure 7.
It must be noted that in this component removing the ambiguity of the word’s meaning is not done directly because some of the words may be mapped to concepts which are more abstract compared to the precise concept of their correspondent, also in this component the precise meaning of a keyword in a specific part of the page is not specified but the important concepts of the page are extracted from the page en masse. The approach of removing the word’s meaning ambiguity is not used in order to increase the efficiency; besides, such an action is not necessary because in constructing the user model, the important concepts of the page are to be considered not the meaning of a specific word in a part of the page. It must be noted that although most of the relationships between the categories in Wikipedia are of is-a type, Wikipedia hasn’t explicitly specified the type of this relationship in its dump structure.

**Studying the suggested plan’s accordance with the question**

The model which is obtained from the user through this method will qualify all the previously mentioned conditions:

- By improving the method, the keywords vector has been obtained.
- It is implicit because it needs only the web server logs.
- The web usage information required for the model has been obtained from the user’s behavior in a specific time period in the website.
- As previously mentioned, all the system components act without human interference.
- It has a good coverage of the domain specific concepts and nominated entities since it exploits Wikipedia for extracting the concepts. Also the weight of each concept has a direct proportion with the number of occurrences of its relevant keywords in the page and in the time period spent by the user in that page, thus it reflects the user’s interest.
- The semantic relationships have been used based on both gloss and hierarchy to improve the model. In the domain concepts extraction component, filtering the concepts is done based on the semantic similarity. In the mapping component, particularly, the hierarchical relationships in the hierarchical structure of Wikipedia are used for mapping the keywords into the concepts.

```
Function extractConceptVector(User u)
    For each keyword k in session pages
        For each concept c in filteredConcepts
            Calculate relatedness_{CF}(k, c);
        End for
    Assign concept with maximum relatedness to k;
    End for
    For each concept c in filteredConcepts

Figure 7: pseudo-code of the mapping component
```
It was observed that the processes of the domain concepts extraction, domain concepts filtering, keywords extraction, and mapping them into the domain concepts in the suggested system are performed automatically. Some of the features of this system which discriminates it from the previous ones include: using a highly rich semantic source for improving the model; automatically extracting the domain specific concepts; automatically filtrating the domain concepts related to the website using the website’s directory structure and automatically mapping the keywords into the domain concepts. Furthermore, construction of the model doesn’t decrease the efficiency of the personalization system due to the system’s offline performance.

Log primary preprocess component
Web server log reception method
The web servers have the capability to create the log files with specific commands. In this section we explain the method of receiving the log in two well-known web servers, apache and IIS:

- Receiving the log in apache: the location and content of the access log are controlled by the CustomLog command. The LogFormat command can be used to simplify selecting the logs’ content. The CustomLog command has the following features:

Table 1: CustomLog command structure

<table>
<thead>
<tr>
<th>Description</th>
<th>Sets filename and format of log file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>CustomLog file</td>
</tr>
<tr>
<td>Context</td>
<td>server config, virtual host</td>
</tr>
<tr>
<td>Status</td>
<td>Base</td>
</tr>
<tr>
<td>Module</td>
<td>mod log config</td>
</tr>
</tbody>
</table>

The LogFormat command has the following structure:

Table 2: LogFormat command structure

| Description | Describes a format for use in a log file |

Take down details
This section describes the details of taking down the suggested method. To take down the components of the system, the programming language JAVA version 1.5 and Eclipse 3.3 tool have been used. You can see the codes of the components in “Ghaderyan Homepage”. In the following sections we will describe important points about taking down each component of the system.
Receiving the log in ISS: the default location of the log file is in the following path:
c:\winnt\system32\LogFiles\W3SVC1
These files are of ex*.log format. In order to save the logs in your desired location use Internet Service Manager:

2. Windows NT 4.0—Start > Programs > Windows NT 4.0 Option Pack > Microsoft Internet Information Server > Internet Service Manager.

Then you should perform the following steps:
1. Double click on the Internet Information Service option to observe a list of the relevant servers.
2. Double click on the server’s name to observe a list of the sites on the server.
3. Right-click on your intended site’s name and select Properties.
4. Click on “enable logging”. Now you can observe the log file directory field which lists a folder which is now set for receiving log.

For more information refer to ISS.

For taking down this component the WUM software (Spiliopoulou, 1999) has been used which is a useful tool for preprocessing the web server logs. This software has been taken down by JAVA language and is freely accessible. One of the useful features of this software is its user-friendly interface. In the present thesis the users’ sessions have been recognized by this software with a 30-minute threshold time. For the intended analysis, WUM creates a folder and after the primary preprocess of the log file it produce some files as the output; some of these files are Website Visitor in which the features of the users such as ID, IP address, user factor, and cookies information exist, Website Pages in which the accesses pages of the website, their ID, and number of accesses to them exist, and so on. Then the user can launch creating sessions whose consequence is creating a SessionLog file that its default format is as follows:

```
1000011 "1995/08/01/00:01:32" "[5000000;1]" "[3000017;1]" "[3000023;1]"

1000021 "1995/08/01/00:03:45" "[5000000;1]" "[3000001;1]" "[3000049;1]"

```

The entries of each line include: session ID, session start date, session duration (time length), and accessed pages ID. There are some setting options for entering the time of observing the page and other required information. This file is given, in the log primary preprocess component, as input and the efficient data structure is build based on it.

Component of extracting the keywords from the website pages

Accessing the Wikipedia data doesn’t require the crawling. Actually this site updates the dumps of the XML related to the site every week and provides them for downloading and recommends the users to use these dumps instead of crawling (which results in the web server’s overload). Wikipedia’s dumps are accessible for various information such as articles, users’ comments and ideas, and so on. The important dump used in the present thesis is articles.xml. The main problem of using Wikipedia for research projects is that the volume of its dumps is too much, about several gigabytes, and since they are as XML they result in a big challenge for its optimum usage; furthermore, since the present project needs various lexical semantic information, it requires a high volume of programming. Therefore in this thesis, after investigating different tools, the JWPL tool (Zesch, 2008) was chosen for achieving more access to the Wikipedia data.

JWPL (JAVA Wikipedia Library) is an application (applied) programming interface with JAVA language which enables access to all the information in Wikipedia including references, classification (categorization), articles, link structure, etc., which has been taken down for high scale NLP programs. One of the advantages of this API which facilitates its usage is abstracting the data sources (data bases) of Wikipedia in an objective format. For example, there are numerous classes in it such as category, page, title, etc. JWPL has been designed based on an object-relational model and, first, creates a database in Mysql from the XML file and, then, performs the accesses through this database.

In the structure of the dumps of the articles, the links are presented in brackets. If in a dump the brackets appear as [ab] this means that in this section there is a link to an article, whose topic is “a”, which is observed in the HTML file with label “b”. If the topic of the referred article is the same as its label in the referring article, it will appear in the dump as [[a]]. For example, if in the XML file the sentence “Shakespeare is buried in the [chancel] of [Holy Trinity Church, Stratford-upon-Avon|Holy
Trinity Church] in [[Stratford-upon-Avon]]" has appeared this means that there are three links to the article in this sentence among which Holy Trinity Church is a label for an article whose topic is Holy Trinity Church, Stratford-upon-Avon.

Although JWPL is a useful tool for working with Wikipedia, it lacks some of the requirements of the keywords extraction component as default. For instance, for finding the number of occurrences of a word in the articles we should, by ourselves, take down this capability from the Page class using the GetText() method which converts the XML text of a page to a simple text.

Before extracting the keywords from the content of the website pages, all the HTML tags were deleted using HTML Parser which is a JAVA powerful and open source library for working with web pages. Also, in order to extract the n-grams from the website pages the opennlp library has been used.

Component of extracting the domain concepts from the website
JWPL has not taken down the calculation of the semantic relationship between two words. Thus the approaches used for calculating the semantic relationship in this component were taken down using Category, and Page classes from JWPL. Also the keywords of the pages’ addresses were extracted from the Website Pages file which is one of the outputs of the WUM software.

Keywords vector constructor component
The code of this component was written in JAVA language. In order to take down the tables and matrices, the JAVA HashMap structures, which are very efficient, and the Serialization capability were used.

Mapping component
In order to take down this component, the wap semantic relationship criterion was taken down using methods of Page and Category classes.

Performing tests
The results of testing with Wikipedia should be compared with the results of testing with WordNet. In this way, the following steps were performed in the keywords extraction component:

- Eliminating the HTML tags
- Eliminating the stopwords (words which don’t give important information about the documents’ content, such as the, a, is, etc.)
- Stemming the words using the Porter algorithm (Porter, 1980)
- Indexing the documents
- Calculating tf-idf
- Extracting the keywords

Version 2.0 of WordNet was selected for taking down. For accessing to WordNet the well-known WordNetSimilarity module, which is in Perl language and can be downloaded from “WordNetSimilarity", was used.

Taking down is a very important step in performing a research project and makes the research tangible for the researcher. Lack of enough knowledge about the appropriate and accepted tools and instruments is one of the important obstacles which can lead to the project failing or non-credible and unreliable results. In this chapter the tools used for taking down the thesis were introduced. Using these tools result in a notable thrift in time and activities required for taking down. For newer versions of these tools you can refer to the mentioned references.

Suggested method evaluation
In this section the tests done for evaluating the suggested methods are presented. First, the used dataset and its features are investigated and, then, the evaluation parameters are introduced. Then the tests and their results are reported. At the end, the tests results are analyzed.

Dataset
One of the main difficulties of researching in the web personalization field is the shortage of standard datasets. Due to the privacy issue, the web server’s logs usually are not brought to public access. This issue becomes more difficult when the information of the website’s content is required. All the existing datasets belong to past years and their referenced pages, if still exist, have certainly been changed too much and thus they are not appropriate for evaluation (e.g. Depaul CTI data & Perkowitz data).

As for the datasets there are two notable points to be mentioned. The first point is related to the number and content of the pages. The pages don’t exist for each lesson and the existing pages don’t have desirable quality because the concepts related to the lessons are not observed in these pages. The second point is related to the number references to these pages. Generally the number of the individuals referring to these pages and the number of their accesses are low and thus the logs of these pages are not appropriate statistical societies for evaluation.

Furthermore, in all the articles which have used the web pages’ content for improving the user model, the data is related to the website of the university and college of the article writers, or an e-commerce website for online sales, and thus this data is not publicly accessible.

The pages of a college website have been used for evaluation. The logs of the web server of the Computer Engineering College of Sharif Polytechnic University have been used for testing. This dataset includes about 6000 web pages about different courses of various school years and terms together with the personal pages of the professors, students, and so on. The log were collected for one
week from 2008 April 13th to 2008 April 20th and after deleting the personal information about 264883 access to these pages were extracted. In the tests the Wikipedia dump related to 2007 February has been used. The public category of the tested website is Computing. After the first searching of this category’s level about 25211 primary concepts were obtained which, after filtering by the keywords of the URLs, 19102 relevant concepts remained. After extracting all the categorized articles in the primary categories, 419937 articles were obtained which created the primary glossary and after adding the Surface Form to it, the size of this glossary set reached to 429694.

Evaluation parameters
The evaluation method used in this project is the same as the method used in all other articles in this field. The final output of the suggested system, that is the concepts vector, is evaluated. Therefore, 100 sessions are randomly chosen and then mapped into the concepts vector once by human and the next time by the system. It must be noted that in mapping the sessions into the concepts by human the pages are investigated by him and then he, himself, extracts the concepts. Then these concepts are compared with the system’s output concepts. In order to simulating the human’s behavior, all the concepts whose weight is less than the concept with highest weight, as much as a fixed difference (20%), are eliminated. So the whole performance of the system is evaluated not merely the mapping component. The sessions are individually (one by one) by the system because in the given dataset it is not possible to recognize the users individually; besides, this does not disarrange the process since the concepts weights are obtained from their average weight in all of the user’s sessions, thus if mapping the sessions into the concepts by human the pages are investigated by him and then he, himself, extracts the concepts. Then these concepts are compared with the system’s output concepts. In other words, what is important is correctly mapping the sessions into the concepts.

The evaluation parameters used in this project are: precision and recall. If one session of the user is mapped, by human, into the concepts set \( H \) and, by system, into concepts set \( S \) then the intended values are calculated as follows:

**Precision:** includes the proportion of the number of the system’s output concepts which are relevant.

\[
\text{precision} = \frac{|H \cap S|}{|S|}
\]

**Recall:** includes the proportion of the relevant concepts which exist in the system’s output.

\[
\text{recall} = \frac{|H \cap S|}{|H|}
\]

**Investigating the system components correctness**
In order to investigate the log primary preprocess component, the reader can refer to Spiliopoulou, 1999. Evaluation of extracting the keywords from the website pages especially important and is explained in Mihalcea, 2007. The outputs of the component of extracting the domain concepts from the website and the keywords vector constructor component are not evaluated directly but their performances are evaluated together with the mapping component performance wholly in the system’s output format.

**Tests**
Three tests have been performed. The first test is related to the suggested system with Wikipedia source. The second test has been performed through substituting Wikipedia by WordNet and some changes have been exerted on the components’ performance. WordNet, in case of lacking lexicology, is considered as a state-of-the-art semantic source. The third test is related to the well-known method based on the tf-idf keywords. Indeed, the suggested model (ignoring its semantic source) is evaluated in 1st and 2nd tests and comparing its results with the 3rd test’s results is the main goal of evaluation in the present thesis. Comparing the 1st and 2nd tests’ results can be considered actually as the comparison of WordNet and Wikipedia semantic sources.

**Hardware**
These tests have been performed using a Dual-Core Pentium processor with 3.4GH frequency, a 2GB RAM, and WINDOWS XP version SP2-2002. The details of taking down process are explained in chapter 5.

**Test results**
Table 3 shows the results of three tests. The average precision value and average recall value have been obtained after calculating the precision and recall for each of the 100 sessions and averaging.

**Table 3: results of tests on 100 random sessions**

<table>
<thead>
<tr>
<th>Test</th>
<th>Average Recall</th>
<th>Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51.1%</td>
<td>49.6%</td>
</tr>
<tr>
<td>2</td>
<td>42.5%</td>
<td>37.8%</td>
</tr>
<tr>
<td>3</td>
<td>31.1%</td>
<td>29.5%</td>
</tr>
</tbody>
</table>

**Test results analysis**
Comparing the results of the two first tests with the third test demonstrate that using the semantic relationships in the web personalization process leads to increase of the user model precision which is not considered in the keywords approach. The reason for low precision of the keywords approach is that the number of a word’s repetition is not a proper measure to evaluate its importance because a word may be repeated many times but have no importance in the domain. Furthermore, a concept
with high importance may appear in various forms in the web pages, also it may have low occurrences in the pages which results in a low recall value for the keywords method. Besides, comparing the results of the two first tests demonstrates the Wikipedia’s advantage to WordNet. The Wikipedia’s higher precision and recall, compared to WordNet, is resulted from its good coverage of domain specific concepts and nominated entities. For example, many of the concepts which exist in the tested website pages such as VLSI, digital circuit, software engineering, etc., don’t exist in WordNet and thus have lower recall; while, there perfect articles in Wikipedia for each of these concepts. Generality of the concepts in WordNet causes that some general concepts are selected as important concepts and, consequently, this leads to lower precision.

Another point which can be deduced from the first test results is that this method is highly dependent on the recall and precision of the keywords extraction component and it seems that this component plays an important role in the system efficiency.

It must be noted that the tested website’s content has had direct effects on the obtained values. By studying the pages we can find out that although there are many “computer science” domain specific concepts (mostly absent in WordNet) but limited information about the lessons’ content is observed in the lessons’ pages and it seems that if the 1st test is performed on the access logs of the websites with high quality content, the position and merits of Wikipedia will be manifested more suitably. But unfortunately, as mentioned before, the websites’ access logs are not publicly accessible.

This section presented the evaluation results which demonstrate the advantage of the suggested method in using semantics for user modeling either by WordNet or by Wikipedia compared to the keywords method. Also, based on the evaluation results, the advantage of Wikipedia compared to WordNet in improving the user model was proved. The 12% increase in precision and 9% increase in recall indicate that Wikipedia can be considered as an effective semantic source in the web personalization process.

References


