Available online at www.sciencedirect.com
ScienceDirect
Knowledge-Based Systems xxx (2006) xxx–xxx
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

A Knowledge Base for the maintenance of knowledge extracted from web data

Juan D. Velásquez [a,*], Vasile Palade [b]

[a] Department of Industrial Engineering, University of Chile, República 701, Santiago, Chile
[b] Computing Laboratory, University of Oxford, Parks Road, Oxford OX1 3QD, UK

Received 20 July 2005; accepted 3 May 2006

Abstract

By applying web mining tools, significant patterns about the visitor behavior can be extracted from data originated in web sites. Supported by a domain expert, the patterns are validated or rejected and rules about how to use the patterns are created. This results in discovering new knowledge about the visitor behavior to the web site. But, due to frequent changes in the visitor's interests, as well as in the web site itself, the discovered knowledge may become obsolete in a short period of time. In this paper, we introduce a Knowledge Base (KB), which consists of a database-type repository for maintaining the patterns, and rules, as an independent program that consults the pattern repository. Using the proposed architecture, an artificial system or a human user can consult the KB in order to improve the relation between the web site and its visitors. The proposed structure was tested using data from a Chilean virtual bank, which proved the effectiveness of our approach.

Keywords: Knowledge Base; Web data; User behavior

1. Introduction

During the last years, we have witnessed a tremendous growth of business performed via the Internet. This led to an explosion in the number of Internet sites and also in the number of visits to these sites.

Web usage mining (WUM) has contributed to the analysis and exploration of the information that visitors leave behind after navigating through a particular web site. Many algorithms and systems have already been proposed for this purpose [14,13,15,18,24].

By applying these techniques, significant patterns about the visitor behavior and his/her preferences can be discovered. The validation process can be performed by an expert in the particular business domain. Additionally, the expert could give some recommendations about how to use the discovered patterns [19].

In this paper, a Knowledge Base (**KB**) for storing the visitor behavior patterns extracted from web data is introduced [21]. By using a data warehouse architecture [3,4,11], two repositories to store the information and the knowledge extracted from web data are defined. In the first repository, information taken from web logs and web pages is stored. The discovered knowledge needs a more complex repository to be used. With this aim in mind, a KB [7] composed of a Pattern Repository and a Rule Repository (containing rules about how to use the patterns) is introduced. Both repositories represent the source used by different types of knowledge users to perform a navigation or web site modification recommendations.

The knowledge contained in the KB can be used by a human being or an artificial system, like an intelligent web site [16,20], in order to improve the relationship with a prospective visitor of the site.

* Corresponding author.
E-mail addresses: jvelasqu@dii.uchile.cl (J.D. Velásquez), vasile.palade@comlab.ox.ac.uk (V. Palade).

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2006.05.015

This paper is organized as follows. Section 2 provides a short discussion about knowledge representation. The methodology to extract significant patterns from web data is introduced in Section 3. In Section 4, the knowledge to be represented and stored is introduced. The framework to acquire and maintain information and knowledge extracted from web data is explained in Section 5. In order to test the effectiveness of the proposed methodology, a real world experiment is performed and shown in Section 6. Finally, Section 7 contains some conclusions.

2. Knowledge representation

A web mining tool developed for a particular web site allows discovering significant patterns about the visitor behavior and his/her preferences. However, the collaboration of an expert is required to validate the patterns and give a short description about how to use them [19].

This approach has pros and cons, for instance, a problem may appear if the expert decides to leave the institution that owns the web site. Usually, the expert will take the expertize with him. The above scenario is a motivation for finding a method to efficiently represent knowledge [8], i.e., to express in some form what we know about a specific subject.

The Knowledge Representation (KR) is a first step in developing an automatic system that uses the knowledge discovered from web data to perform actions. Finding a proper method of knowledge representation is not a trivial task. Fundamentally, the knowledge representation describes "how an entity sees the world", in order to understand a situation and prepare a correct action for that situation. In other words, it provides the capacity to infer new expressions from old ones.

2.1. Rules

Following the knowledge representation as ontological commitments, a set of rules about how to use the discovered patterns can be defined [5]. The rules often specify recommendations, directives and strategies. In a computational form, they are expressed as instructions ***If*** < *condition* > ***Then*** < *recommendation*>.

These expressions allow to easily represent the expert knowledge. However, when the set of rules grows, it is difficult to specify which rule is the most appropriate to be applied.

The rules associate facts with actions (recommendations) through matching facts and conditions, as shown in Fig. 1.

In this example, if the visitor visits the page p_1 and spends time t_1 on it, then the recommendation is "go to page p_{10}". Also, if the visitor browsing behavior belongs to cluster c_1, then the recommended pages to visit are p_3,p_5,p_8.

2.2. Knowledge repository

The problem is now how to maintain the discovered knowledge. A good approach is by storing it in a repository, following the same method used for data. However, the knowledge is more complex than just simple data. It corresponds to the patterns discovered after processing data, which are translated into rules on how to use the patterns.

R_1: If VisitPage(p_1) and SpentTime(t_1) Then RecommendationPage(p_{10})

R_2: If BelongCluster(c_1) Then RecommendationPage(p_3, p_5, p_8)

R_3: If CountPageVisit(p_i) < D Then DeletPage(p_i)

... ...

R_n: If ...

Fig. 1. Knowledge representation using rules.

The KB is a general structure for storing facts (patterns) and rules about their use. Its typical representation corresponds to keeping track of rules that share common wisdom [7].

In a practical realization, the KB must be able to maintain rules in an easy way. This becomes a complex task when the problem conditions change in time, as it is the case with the knowledge extracted from web data.

3. Visitor behavior patterns extracted from web data

In recent works [6,17,24], web mining tools have been applied on data originated in a web site in order to understand the visitor behavior. In this sense, clustering techniques have considerably contributed to extracting significant patterns about the visitor browsing behavior and visitor text preferences [25].

Before applying web mining techniques, the data are transformed into behavior patterns, using a specific model about the visitor behavior.

3.1. Preprocessing web logs

The task is to determine, for each visitor, the sequence of web pages visited during a session, based on the available web log files. This process is known as **sessionization** [1]. A maximum time duration of 30 min per session is considered. The transactions that belong to a specific session can be identified using tables and program filters. We consider only web log registers with non-error codes, whose URL parameters link to web page objects.

3.2. Preprocessing of the web site

The web site is represented by a vector space model [2]. Let R be the number of different words in a web site and Q the number of web pages. A vectorial representation of the web site is a matrix M of dimension RxQ, $M = (m_{ij})$ where $i = 1,\ldots,R, j = 1,\ldots,Q$ and m_{ij} is the weight of the ith word

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

in the jth page. To calculate these weights, we use a variant of the *tfxidf-weighting*, defined as follows:

$$m_{ij} = f_{ij}(1 + sw(i)) * \log(\frac{Q}{n_i}) \quad (1)$$

where f_{ij} is the number of occurrences of the ith word in the jth page, $sw(i)$ is a factor to increase the importance of special words and n_i is the number of documents containing the ith word. A word is special if it shows special characteristics, e.g., the visitor searches for this word.

Definition 1 (*Page vector*)**.** It is a vector $\mathbf{WP^j} = (wp_1^j, \ldots, wp_R^j) = (m_{1j}, \ldots, m_{Rj})$ with $j = 1, \ldots, Q$, that represent a list of words contained within a web page.

It represents the jth page by the weights of the words contained in it, i.e., by the jth column of M. The angle's cosine is used as a similarity measure between two page vectors:

$$dp(WP^i, WP^j) = \frac{\sum_{k=1}^{R} wp_k^i wp_k^j}{\sqrt{\sum_{k=1}^{R}(wp_k^i)^2}\sqrt{\sum_{k=1}^{R}(wp_k^j)^2}}. \quad (2)$$

3.3. Modeling the visitor browsing behavior

Our visitor behavior model uses three variables: the sequence of visited pages, their contents and the time spent on each page. The model is based on a n-dimensional visitor behavior vector which is defined as follows.

Definition 2 (*Visitor behavior vector*)**.** It is a vector $\upsilon = [(p_1,t_1)\ldots(p_n,t_n)]$, where the pair (p_i,t_i) represents the ith page visited (p_i) and the percentage of time spent on it within a session (t_i), respectively.

3.4. Comparing visitor sessions

Let α and β be two visitor behavior vectors of dimension C^α and C^β, respectively. Let $\Gamma(\cdot)$ be a function that returns the navigation sequence corresponding to a visitor vector. A similarity measure has been proposed elsewhere to compare visitor sessions, as follows [24]:

$$sm(\alpha,\beta) = dG(\Gamma(\alpha),\Gamma(\beta))\frac{1}{\eta}\sum_{k=1}^{\eta}\tau_k * dp(p_{\alpha,k}, p_{\beta,k}) \quad (3)$$

where $\eta = \min\{C^\alpha, C^\beta\}$, and $dp(p_{\alpha,k},p_{\beta,k})$ is the similarity (Eq. 2) between the kth page of vector α and the kth page of vector β. The term $\tau_k = \min\{\frac{t_{\alpha,k}}{t_{\beta,k}}, \frac{t_{\beta,k}}{t_{\alpha,k}}\}$ is an indicator of the visitor's interest in the visited pages. The term dG is the similarity between sequences of pages visited by two visitors [17].

3.5. Modeling the visitor's text preferences

A web site keyword is defined as *a word or a set of words that makes the web page more attractive to the visitor* [22]. The task here is to identify which are the most important words (keywords) in a web site from the visitor's viewpoint. This is done by combining usage information with the web page content and by analyzing the visitor behavior in the web site.

To select the most important pages, it is assumed that the degree of importance is correlated with the percentage of time spent on each page within a session. By sorting the visitor behavior vector according to the percentage of time spent on each page, the first ι pages will correspond to the ι most important pages.

Definition 3 (ι – Most important pages vector)**.** It is a vector $\vartheta_\iota(\upsilon) = [(\rho_1,\tau_1),\ldots,(\rho_\iota,\tau_\iota)]$, where the pair (ρ_ι,τ_ι) represents the ιth most important page and the percentage of time spent on it within a session.

Let α and β be two visitor behavior vectors. A similarity measure between two ι – most important pages vectors is defined as:

$$st(\vartheta_\iota(\alpha),\vartheta_\iota(\beta)) = \frac{1}{\iota}\sum_{k=1}^{\iota}\min\{\frac{\tau_k^\alpha}{\tau_k^\beta},\frac{\tau_k^\beta}{\tau_k^\alpha}\} * dp(\rho_k^\alpha,\rho_k^\beta) \quad (4)$$

where the term $\min\{\cdot,\cdot\}$ indicates the visitors' interest in the visited pages, and the term dp is the similarity measure (Eq. 2).

In Eq. (4), the content similarity of the most important pages is multiplied by the ratio of the percentage of time spent on each page by visitors α and β. This allows us to distinguish between pages with similar contents, but corresponding to different visitors' interests.

3.6. Applying web mining techniques

Similar visitor behaviors are grouped into clusters with common characteristics, such as the navigation sequence or the preferred web pages.

3.6.1. Clustering the visitor sessions

For clustering the visitor sessions, a Self-Organizing Feature Map (SOFM) [12,25] was applied using the similarity measure in Eq. (3). The SOFM requires vectors of the same dimension. Let H be the dimension of the visitor behavior vector. If a visitor session has less than H elements, the missing components up to H are filled with zeroes. Otherwise, if the number of elements is greater than H, only the first H components are considered.

3.6.2. Clustering the ι-most important pages vectors

A SOFM is used to find groups of similar visitor sessions. The most important words for each cluster are determined by identifying the cluster centroids. The importance of each word with respect to every cluster is calculated by:

$$kw[i] = \sqrt[\iota]{\prod_{p\in\zeta} m_{ip}} \quad (5)$$

for $i = 1,\ldots,R$, where kw is an array containing the geometric mean of the weights of each word (Eq. 1) within

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

the pages contained in a given cluster. Here, ζ is the set of pages contained in the cluster. By sorting kw in descending order, the most important words for each cluster can be selected.

4. Representation of the extracted knowledge

The patterns discovered after applying the web mining tools correspond to the cluster centroids extracted from the visitor behavior vectors and most important page vectors.

The cluster interpretation is a subjective task [19]. While for some persons a cluster may not make much sense, others discover new knowledge in it. That's why it is convenient to be assisted by a business expert for a relatively good interpretation, and see "how to use the patterns found". Because the usage of this new knowledge could result in different actions, it is convenient to focus on a selected group of them. In our case, we are interested in support recommendations for the web site structure and content modifications.

By analyzing the first group of centroids that correspond to the visitor browsing behavior, two types of recommendations could be implemented:

- Online recommendations. Given a new visitor and his/her behavior vector, it is possible to construct an online vectorial representation about the visited pages and the time spent on each of them. Next, the visitor behavior is classified by selecting the nearest centroid using the similarity measure in Eq. (3). From the information contained in the centroid, a prediction about which would be the most interesting pages for the visitor can be made.
- Offline recommendations. These correspond to structure and content changes proposed by the web master. The centroids are a summary of the typical visitor behavior, i.e., they show what pages have been searched. For example, a page could be erroneously placed in the web site, making difficult for visitors to find it. Then, a structural change about modifying the corresponding links could be suggested.

In order to prepare good recommendations, it is important to take into account the statistics on the web page access.

By analyzing the second group of centroids, corresponding to text preferences, the more significant words for the visitor are extracted. For the moment, only offline recommendations about the web site content can be made. The keywords can be used as:

- Link words: Typical words that have a link to a web page.
- Marked words: Words marked with different colors to display the importance of some concepts.
- Searching words: Several search engines, like google, yahoo, altavista, etc., have the option to customize the storage of the web site. The web site owner (or web master) will specifically want that the search engine crawler rescue the complete web site, and index its content by giving special attention to a set of words. Then, when a visitor is looking for a specific page that contains some words of his/her interest, the search engine can come up with the web site pages in a more straightforward manner.

Any representation of the knowledge described above must consider that:

- Different visitors have distinct goals;
- The behavior of a visitor changes over time;
- A site tends to grow in time by accumulating many pages and links; without being restructured according to new needs.

5. A framework to knowledge discovery from web data

By representing patterns and recommendations as rules could result in generating a large set of rules. Due to frequent changes in the visitor's interest and the web site itself, the recommendations might become obsolete in a short period of time.

In this paper, we propose to maintain the patterns by storing them in a database-like repository, and the rules as an independent program that consult the patterns repository when preparing the recommendations. Because the repository will contain patterns discovered in different time periods, it is convenient to apply the data mart technology. Also, it is necessary to develop generic parametric rules.

5.1. Overview

In Fig. 2, a framework to acquire, maintain and use knowledge about the visitor behavior is shown [21]. The main idea is to use an Information Repository to store the data to be analyzed, and a Knowledge Base to contain the results from these analyses, as well as additional domain knowledge from expert. This allows to suggest online navigation steps, and change the web site structure and contents offline.

The respective data sources are web log registers (dynamic information) and the text contained in the web pages (static information). The former are based on the structure given by W3C. The latter is the web site itself, i.e., a set of web pages with a logical structure. A preprocessing stage is undertaken and the relevant information is loaded onto the Information Repository.

By applying data mining techniques on the web data, it is possible to find unknown and hidden knowledge [23,21]. This task is performed using an automatic algorithm that directly interacts with the web data.

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

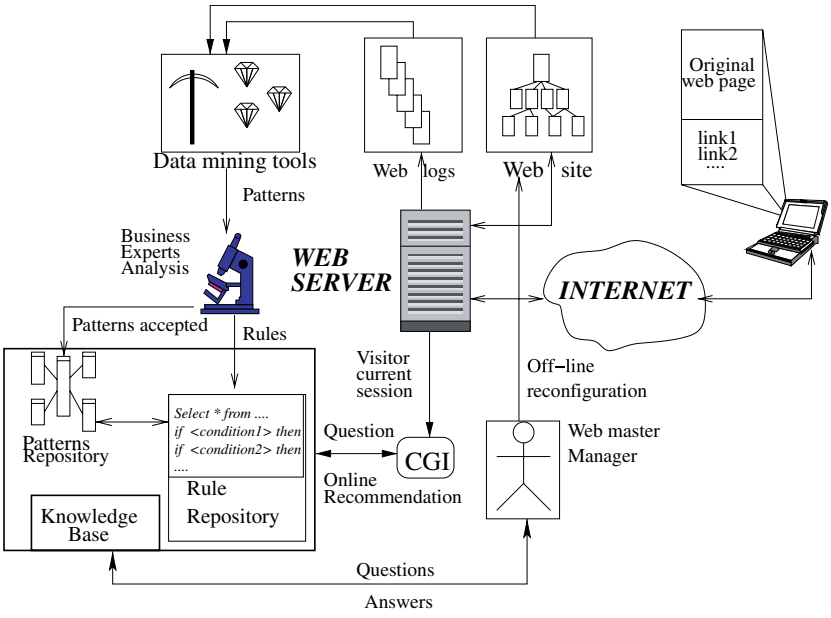

Fig. 2. A framework to discover and use knowledge extracted from web data.

The patterns extracted by the data mining tools must be validated by a business expert and finally loaded onto the Pattern Repository. The specific usage of the patterns is implemented within rules, which are loaded onto the Rule Repository.

Pattern and rule repositories form the complete structure of the Knowledge Base, which is used to suggest different kinds of modifications in the web site. Because we are working with historical repositories, a way to estimate the future success of the proposed changes is to check what happened with the visitor behavior, when similar changes had been made in the past.

Making online changes in the visitor navigation could determine the visitors reject the changes for considering them too invasive. The same situation could happen for changes performed offline, directly following the information and knowledge inside the data marts. For many reasons, it is more convenient to give only recommendations about the changes in the structure and content of the web site, either online or offline [10]. The visitors, or the web master, should decide whether actually make the changes or not.

The approach introduced here has two kinds of potential ''knowledge users'': human beings and artificial systems. In the first case, the human beings consult both repositories as a Decision Support System and propose changes in the web site. These changes are usually performed by hand, although part of them can be automated. In the second case, the artificial systems use mainly the pattern repository and return navigation recommendations as a set of links to web pages. Then, dynamic web pages can incorporate these links in the information which will be sent to visitors.

In the next subsections, each element of the framework proposed above will be explained in detail.

5.2. Knowledge users

The proposed framework here takes into account two kinds of knowledge users: an automatic system, e.g., an inference engine, or a human user.

In the case of automatic systems, the aim is to prepare an online navigation recommendation. In this case, an interface between the web server and the Knowledge Base is required. Notice that the answer for the visitor must be sent in HTML standard, in order to avoid interpretation problems with the web browser.

Using the Common Gateway Interface (CGI) specifications, the web server can interact with an external system, as shown in Fig. 2. A program developed in any language that support standard I/O and has a database connection interface can consult the KB and prepare a HTML code in order to include additional information in the original web page requested by the visitor.

The Knowledge Base can also be consulted by a human expert, for example a web master or a marketing manager, and offline changes can be recommended with respect to the web site structure and content.

5.3. Knowledge Base

Using own expertise, a domain expert can interpret these patterns and build rules for a given task, in our case for online navigation suggestions.

The Knowledge Base [5] implements wisdom representation through the use of ''if-then-else'' rules based on the discovered patterns. Fig. 3 shows the structure of the proposed Knowledge Base. It is composed by a Pattern Repository, where the discovered patterns are stored, and a Rule Repository, which contains the general rules about how to use the patterns.

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

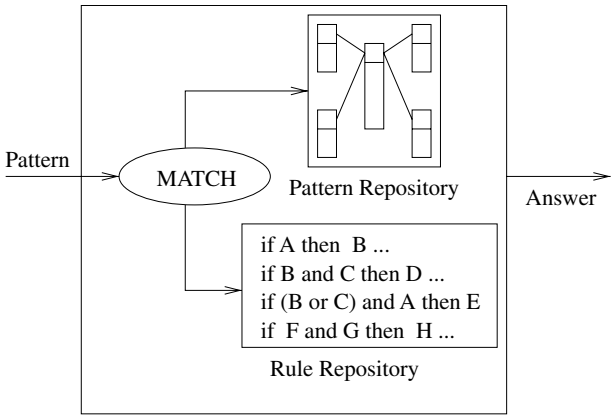

Fig. 3. The proposed Knowledge Base structure.

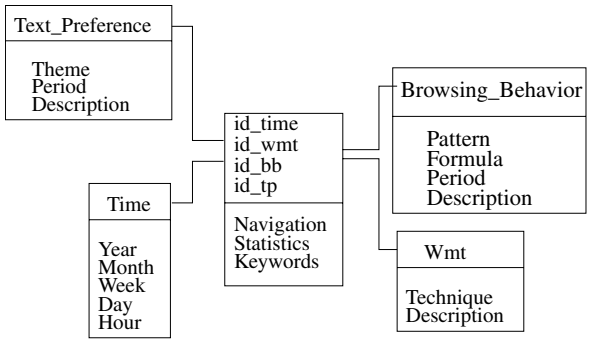

Fig. 4. The proposed Pattern Repository model.

In order to use the KB, when a pattern is presented, a matching process is performed to find the most similar pattern in the Pattern Repository. With this information available in the Rule Repository, the set of rules that will create the suggestion of the KB is then selected.

5.3.1. Pattern repository

In the literature, only web access data is stored, see for example [3,11,23]. We propose to store the discovered patterns.

The Pattern Repository stores the patterns revealed from the Information Repository by applying web mining techniques. Fig. 4 shows a generic model of the Pattern Repository, which is based on Data Mart technology.

The pattern extraction process uses a matching function (column **formula** in table **browsing_behavior**) to find the most similar patterns, within the Pattern Repository, to the sample presented to the system. This repository can also be implemented using the data mart architecture in star model [9]. In the fact table shown in the middle of Fig. 4, the measures are **navigation**, **statistics** and **keywords**. These measures are non-additives [11] and contain the Web page navigation suggestions, related statistics, such as the percentage of visits in the period of study and the keywords discovered. The dimensional table **time** contains the date of the application of the Web mining technique over the information repository. The **browsing_behavior** table contains the patterns found about the visitor browsing behavior. In the *formula* column, the specific expression used for the feature vector comparison is stored, and the *description* column contains the details. The *period* column contains the period of time when the data to be analyzed was generated, e.g., "01-Jan-2003 to 31-Mar-2003". The **text_preferences** table contains, in the *theme* column, a context description of the web site keywords. For instance, a context for keywords related to credit cards is "promotion about VISA credit card". The table **wmt** (Web mining technique) stores the applied mining technique, e.g., "Self-Organizing Feature Map (SOFM)", "K-means", etc.

When the **KB** is consulted, the Pattern Repository returns a set of possible web pages to be suggested. Based on this set and additional information, such as statistics of the accessed web pages, the Rule Repository makes the final recommendations.

5.3.2. Rule repository

The goal of applying the Rule Repository is to recommend a page from the current web site, i.e., to make a navigation suggestion. Using an online detection mechanism like a cookie, the visited pages and the time spent on them during a session can be obtained.

Using these data, we first match the current visit with the visit patterns stored in the Pattern Repository. This requires a minimum of visited pages to understand the current visitor's behavior.

Then, the Rule Repository is applied to the matching results to give an online navigation suggestion about the next page to be visited, based on the history of previous navigation patterns.

If the suggested page is not directly connected with the current page, but the suggestion is accepted by the visitor, then new knowledge can be generated about his/her preferences. This can be used to reconfigure the web site by reorganizing the links among the pages.

Fig. 5 shows part of the rule set of Rule Repository. The SQL-query extracts, from the Pattern Repository, the patterns to be used in the creation of the recommendation. In this sense, the function "formula" compares the storage patterns with the values originated in the current visitor session. The parameter ϵ is used to identify those patterns that are "close enough" to the current visit, and the parameter δ filters the recommendations whose statistics are above a given threshold, i.e., it is mandatory that the page acquires a minimum percentage of visits.

Since the Pattern Repository contains historical information, a suggested page may not appear in the current web site. In that case, the function "compare_page" determines the page of the current web site, whose content is most similar to that of the suggested page (by using Eq. 2).

The data structure "S" is used to store the query results. It consists of two variables: navigation and statistics, that show the navigation sequence and associate statistics, and are used for preparing the recommendation.

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

```
select navigation, statistics, formula(pattern,n) into S
from pr_fact, time, browsing_behavior, wmt where "star join" and
"fix technique" and "fix time" and formula(pattern,n) > ϵ;
...
  if S is empty then
   send("no suggestion");
...
while S not empty loop
  if S.navigation == compare_page(ws,S.navigation) then
    S.navigation ∉ actual_web_site;
...
  if S.navigation ≠ last_page_visited and S.statics > δ then
   send(S.navigation);
...
end loop
```

Fig. 5. A part of Rule Repository's code.

6. A real world application

We applied the above described methodology to the web site of the first Chilean virtual bank, where all transactions are made using electronic means, like e-mails, portals, etc. (see www.tbanc.cl). We analyzed all the visits done between January and March 2003. Approximately eight millions of raw web log registers were collected. The site had 217 static web pages with texts written in Spanish, which were numbered from 1 to 217, to facilitate the analysis. In Table 1, the web pages are grouped by their main topic.

6.1. Using SOFM for browsing pattern discovery

As mentioned above, the pages in the web site were labelled with a number to facilitate the analysis. Table 1 shows the main content of each page.

The SOFM used has 6 input neurons and 32*32 output neurons in the feature map, with a thoroidal topology.

The cluster identification is performed by using a visualization tool supported by a density cluster matrix, called winner matrix. It contains the number of times the output neurons win, during the training of the SOFM.

Table 1
Bank web site pages and their content

Pages	Content
1	Home page
2,...,65	Products and services
66,...,98	Agreements with other institutions
99,...,115	Remote services
116,...,130	Credit cards
131,...,155	Promotions
156,...,184	Investments
185,...,217	Different kinds of credits

Table 2 shows the clusters found by the SOFM. The second column contains the centroid of the cluster, represented by the sequence of visited pages, and the third column indicates the time spent in each centroid.

A simple cluster analysis shows the following results:

- Cluster 1. Visitors searching information about credit cards.
- Cluster 2. Visitors interested in investments and remote services offered by the bank.
- Cluster 3. Visitors interested in agreements between the bank and other institutions.
- Cluster 4. Visitors that are interested in general products and services offered by the bank.

6.2. How to use the navigation patterns?

The patterns discovered in the clusters analysis serve as a basis for the online and offline recommendations. The recommendations are validated by a business expert. It is also important to use simple statistics to support some of the theories elaborated about the visitor behavior.

Using the discovered clusters, we can classify the visitor browsing behavior into one of them, by comparing the cluster centroid with the current navigation, using the similarity measure introduced in Eq. (3).

The online navigation recommendations are created as follows. Let $\alpha = [(p_1,t_1),\ldots,(p_m,t_m)]$ be the current visitor session and $C_\alpha = [(p_1^\alpha, t_1^\alpha), \ldots, (p_H^\alpha, t_H^\alpha)]$ the centroid such as $\max\{sm(\alpha,C_i)\}$, with C_i the set of centroids discovered and H the dimension of C_i defined in the SOFM. The recommendations are created as a set of pages whose text content is related to p_{m+1}^α. These pages are selected with the expert collaboration.

Let $R_{m+1}(\alpha)$ be the online navigation recommendation for the $(m+1)$th page to be visited by visitor α, where $\delta < m < H$ and δ the minimum number of pages visited to prepare the suggestion. Then, we can write $R_{m+1}(\alpha) = \{l_{m+1,0}^\alpha, \ldots, l_{m+1,j}^\alpha, \ldots, l_{m+1,k}^\alpha\}$, with $l_{m+1,j}^\alpha$ the jth link page suggested for the $(m+1)$th page to be visited by visitor α, and k the maximum number of pages for each suggestion. In this notation, $l_{i+1,0}^\alpha$ represents the "no suggestion" state.

6.3. Online navigation recommendation effectiveness

Usually, the application of any recommendation needs the permission of the web site owner. It is a complicated

Table 2
Visitor behavior clusters

Cluster	Visited pages	Time spent in seconds
1	(162,157,172,114,105,2)	(3,71,112,110,32,3)
2	(1,5,7,10,135,191)	(30,61,160,110,175,31)
3	(72,87,154,188,140,85)	(8,57,31,3,71,91)
4	(110,104,128,126,31,60)	(25,73,42,65,98,15)

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

task due to the fact that sometimes the web site is the core business of the organization and any change could result in a loss in market share. In this sense, we propose a method to test the recommendation effectiveness based on the same web data used in the pattern discovery stage.

In fact, a part of the complete web data is used to extract significant patterns, and for these we define a set of rules. Next, we test the effectiveness with the remaining web data.

Let $ws = \{p_1,\ldots,p_n\}$ be the web site and the pages that compound it. Using the distance introduced in Eq. (2), and with the collaboration of a web site content expert, we can define an equivalence class for pages, where the pages belonging to the same class contain similar information. The classes partition the web site in disjoint subsets of pages.

Let Cl_x be the xth equivalent class for the web site. It is such as $\forall p_z \in Cl_x, p_z \notin Cl_y, x \neq y, \bigcup_{x=1}^{w} Cl_x = ws$ where w is the number of equivalence classes.

Let $\alpha = [(p_1,t_1),\ldots,(p_H,t_H)]$ be a visitor behavior vector from the test set. Based on the first m pages actually visited, the proposed system recommends for the $(m+1)$th page several possibilities, i.e., possible pages to be visited.

We test the effectiveness of the suggestions made for the $(m+1)$th page to be visited by visitor α following this procedure. Let Cl_q be the equivalence class for p_{m+1}; if $\exists l^{\alpha}_{m+1,j} \in R^{\alpha}_{m+1} / l^{\alpha}_{m+1,j} \in Cl_q, j > 0$, then we assume the suggestion was successful.

The number of recommended pages, obtained during the construction of the recommendation, could be large, making the visitor confused about which page to follow next. We set in k the maximum number of pages per recommendation. By using the page distance introduced in Eq. (2), we can extracts the k closest pages to p_{m+1} in the recommendation:

$$E^{k}_{m+1}(\alpha) = \{l^{\alpha}_{m+1,j} \in \mathrm{sort}_k(sp(p_{m+1}, l^{\alpha}_{m+1,j}))\}, \quad (6)$$

with sp the page distance introduced in Eq. (2). The "sort_k" function sorts the result of sp in descending order and extracts the "k" link pages closest to the p_{m+1} page.

A particular case is when $E_{m+1}(\alpha) = \{l^{\alpha}_{m+1,0}\}$, i.e., no suggestion is proposed.

6.4. Web site keywords

By assuming that there is a correlation with the maximum time spent per page in a session, a method to find the web site keywords is introduced along with the Important Page Vector definition.

We set to 3 the maximum dimension of this vector. Then, a SOFM with 3 input neurons and 32 output neurons was used to find clusters of Important Page Vectors.

The neural network training was carried out on a Pentium IV computer, with 1 Gb RAM and running under Linux Operating System, distribution Redhat 8.0. The training time was 25 hours and the number of epochs was set to 100.

Fig. 6 shows, on the x,y axis, the neurons positions in the SOFM. The z axis is the normalized winning frequency of a neuron in the training set.

From Fig. 6, 8 main clusters can be observed, which contain the information about the most important pages in the web site. The cluster centroids are shown in Table 3. The second column contains the center neurons (winner neuron) of each cluster, representing the most important pages visited.

To get the web site keywords, a final operation is required, corresponding to analyzing which words in each cluster have a greater relative importance in the complete web site.

By applying Eq. (5), the keywords and their relative importance in each cluster are obtained. For instance, if the cluster is $\zeta = \{7,15,186\}$, then $kw[i] = \sqrt[3]{m_{i7} m_{i15} m_{i186}}$, with $i = 1,\ldots,R$.

Finally, by sorting the kw in descending order, we can select the k most important words for each cluster, for instance $k = 8$.

In Table 4 a selected group of keywords from all clusters is shown. The keywords alone do not make much sense. They need a context in a web page and they could be used as special words, e.g., marked words to emphasize concepts or links to other pages.

The specific recommendation is to use the keywords as "words to write" in a web page, i.e., the paragraphs written in the page should include some keywords, and some of them may be even used as links to other pages.

The keywords could also be used as index words by a search engine, i.e., some keywords could be used in the customization of the crawler that visits the web site and loads the pages. Then, when a user is looking for a specific page in the search engine, the probability of getting the web site is increased.

6.5. Loading the knowledge base

The Knowledge Base presented in Section 5.3 was used to load the patterns and the rules about how to use the pat-

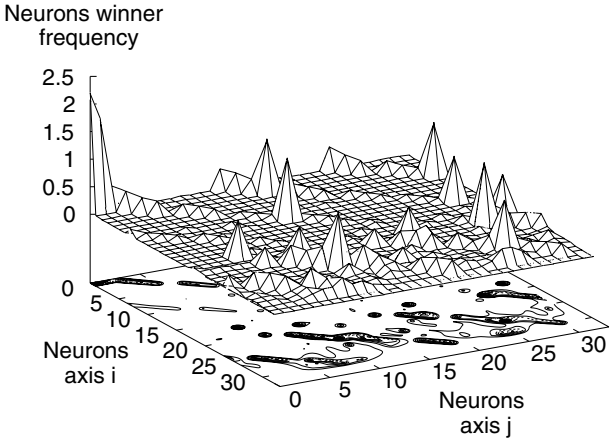

Fig. 6. Identifying clusters of important page vectors.

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

Table 3
Centroids for extraction of keywords

Cluster	Pages visited
1	(161,175,209)
2	(110,130,45)
3	(115,102,1)
4	(161,172,191)
5	(3,9,147)
6	(81,201,144)
7	(7,15,186)
8	(87,178,141)
9	(108,131,62)
10	(91,154,101)

Table 4
Some discovered keywords

No.	Keywords
1	Transferencia
2	Hipotecario
3	Cuenta
4	Promoción
5	Concurso
6	Mercados
7	Crédito
8	Tarjeta

terns in the bank web site. The stored knowledge is mainly used to create online navigation recommendations, but it could be also used to prepare offline recommendations.

6.5.1. Pattern repository

In Fig. 4, the general structure of the pattern repository was presented. The measures in the fact table correspond to the recommended page and to some statistics about its use and the web site keywords for the period analyzed. These patterns are consulted using the information contained in the dimensional tables. An example of this is the following:

- **Time**. (2003, October 4, 26, and 18), i.e., "18:00 h, October 26th, fourth week, year 2003".
- **Browsing_Behavior**. The cluster centroids discovered by the web mining process implied by formula in Eq. (3), and shown in Table 2, as well as with the formula in Eq. (3).
- **Wmt**. Self-Organizing Feature Map with a thoroidal architecture, 32×32 neurons.

The table **Time** shows the date when the web mining tools were applied over the Information Repository. In the dimensional table **Browsing_Behavior**, the information about the clusters centroid is displayed. The **Wmt** table contains information about the specific web mining tool applied.

A human user can apply the results of his/her query in the preparation of an offline structural change recommendation. The same query scheme could be used by an automatic system to prepare online navigation recommendations.

6.5.2. Rules for navigation recommendations

First, we need to identify the current visitor session. Since the selected web site uses cookie, this tool can be used for online session identification.

In order to prepare the online recommendation for the $(m+1)$th page to visit, we compare the current session with the pattern in the Pattern Repository. The comparison needs a minimum of three visited pages ($\delta = 3$) to determine the cluster centroid most similar to the current visit and, in this way, allowing to prepare the recommendation for the fourth page to be visited. This process can be repeated after the visitor has visited more than three pages, i.e., in the recommendations for the fifth page, we use the four pages visited in the current session.

The final online recommendation is made using the developed rule base together with the domain expert. For the four clusters found, sixteen rules were created. We suggest at most three links to follow for the fourth, fifth and sixth pages, i.e., $k = 3$.

6.6. Online and offline recommendations

With the help of a bank's business expert, a list of recommendations was proposed. It includes navigation recommendations, links to be added and/or eliminated from the current site, and words to be used in future pages as content recommendations. Here, only a few recommendations are shown due to a confidentiality agreement with the bank.

Some of this recommendations are currently under evaluation at the bank before their final implementation on the web site.

6.6.1. Structure recommendations

Based on the clustering of similar visits, we made offline recommendations for the reconfiguration of the link structure of the bank web site. Some of these recommendations are:

Add links intra clusters. The idea is to improve the accessibility of pages within each cluster from other pages belonging to the same cluster.
Add links inter clusters. The idea is to improve the accessibility of pages belonging to different clusters that share many common visitors.
Eliminate links. Inter-clusters links that are rarely used can be eliminated.

6.6.2. Content recommendations

The web site keywords represent a set of concepts that could motivate the user interest to visit the web site. Their use as isolated words do not make much sense, since a cluster represents different contexts through a set of keywords. Then, for making recommendations it is straightforward "to use the word in the paragraphs", i.e., if the page writer wants to write about a specific topic, he/she needs to include the web site keywords related to that subject.

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015

6.6.3. Navigation recommendations

The idea is to use the discovered clusters, the statistics associated to each page, and the rules, for creating a correct navigation recommendation.

This process requires on online session identification, therefore a mechanism like a cookie must be implemented in all sessions.

The recommendations are activated when the visitor clicks the third page in a session. The similarity measure is used to define to which cluster belongs the current visitor. The suggested pages appear at the bottom of the selected page.

For instance, if a visitor session matched with the cluster "1" (see Table 3), the most likely pages to be recommended are those relative to Products and Services, Promotions and Credit Cards. Using the statistics related to pages and associated rules, the specific pages to create the recommendations are selected.

6.6.4. Testing the online navigation suggestion effectiveness

Before applying the SOFM on the visitor behavior vectors, it is necessary a final adjustment with respect to the vector size. By construction of the SOFM, the vectors must have the same cardinality. In this case, we consider six components. If a vector does not have at least six elements, it is not considered in our analysis. Otherwise, we only consider up to the sixth component of a session for the visitor behavior vector.

From the 20% of the visitor behavior vectors that belong to the test set, we select only those that have exactly six components. In this way, we have 9751 vectors to test the effectiveness of the online navigation suggestions.

In Fig. 7, the histogram shows the percentage of the accepted recommendations, using the proposed validation method.

If using the proposed methodology, just one page is suggested, slightly more than 50% of the visitors would accept it. This could be considered a very successful suggestion by the business expert, since we are dealing with a complex web site with many pages, many links between pages, and a high rate of visitors that leave the site after few clicks.

Furthermore, it should be mentioned that the percentage of acceptance would probably have been even higher if we actually had suggested the respective page during the session. Since we compared past visits stored in log files, we could only analyze the behavior of visitors that did not actually receive any suggestion we proposed.

7. Conclusions

Using the knowledge extracted from data originated in a web site is a good way to improve the relation with the visitors of that site.

Because the visitor's interest and the web site itself may change in time, it is necessary to develop a flexible structure, able of maintaining the knowledge extracted in the period under analysis. The representation of knowledge as a set of rules is not the best approach, as the number of rules could grow in the time, therefore making difficult their maintenance.

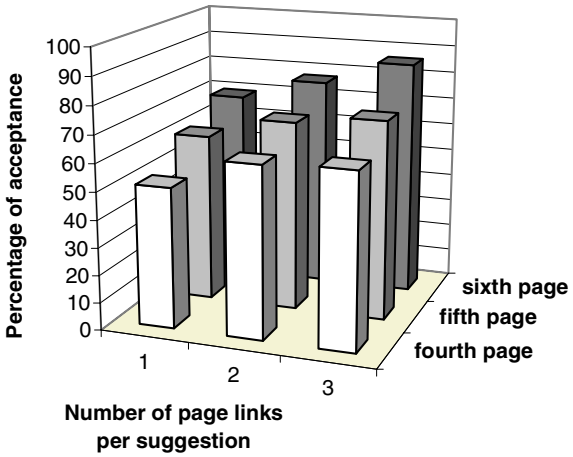

Fig. 7. Percentage of acceptance of online navigation recommendations.

The proposed KB deals with this situation by maintaining two repositories: one for patterns and another one for parametric rules, i.e., the values to evaluate a situation are passed trough parametric variables. This scheme provides a minimum set of rules, an easy maintenance of them and a historic repository of patterns.

In our future work, other web mining techniques will be applied in order to provide new patterns and rules for the KB.

Acknowledgement

The authors thank to the Millennium Scientific Nucleus on Complex Engineering Systems (Chile), which has partially funded this work.

References

[1] B. Berendt, M. Spiliopoulou, Analysis of navigation behavior in web sites integrating multiple information systems, The VLDB Journal 9 (2001) 56–75.
[2] M. Berry, S. Dumais, G. O'Brien, Using linear algebra for intelligent information retrival, SIAM Review 37 (1995) 573–595.
[3] F. Bonchi, F. Giannotti, C. Gozzi, G. Manco, M. Nanni, D. Pedreschi, C. Renso, S. Ruggieri, Web log data warehousing and mining for intelligent web caching, Data and Knowledge Engineering 32 (2) (2001) 165–189.
[4] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, S. Paraboschi, Designing data marts for data warehouses, ACM Transactions on Software Engineering Methodology 4 (2001) 452–483.
[5] M. Cadoli, F.M. Donini, A survey on knowledge compilation, AI Communications 10 (3–4) (1997) 137–150.
[6] R.W. Cooley, Web usage mining: discovery and application of interesting patterns from web data, Dissertation for degree of Doctor of Philosophy. University of Minnesota, Faculty of the Graduate School, Minnesota, USA, 2000.
[7] J. Debenham, Knowledge base maintenance through knowledge representation, in: Proceedings of the 12th International Conference on Database and Expert Systems Applications. München, Germany, September 2001, pp. 599–608.

[8] V. Devedzic, Knowledge discovery and data mining in databases. Tech. rep., School of Business Administration, University of Belgrade, Yugoslavia, 2002.
[9] W.H. Inmon, Building the Data Warehouse, second ed., John Wiley and Sons, New York, 1996.
[10] M. Kilfoil, A. Ghorbani, W. Xing, Z. Lei, J. Lu, J. Zhang, X. Xu, Toward an adaptive web: the state of the art and science, in: Proceedings of the Annual Conference on Communication Networks and Services Research. Moncton, Canada, May 2003, pp. 119–130.
[11] R. Kimball, R. Merx, The Data Webhouse Toolkit, Wiley Computer Publisher, New York, 2000.
[12] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag, 1987.
[13] R. Kosala, H. Blockeel, Web mining research: a survey, SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining 2 (1) (2000) 1–15.
[14] Z. Lu, Y. Yao, N. Zhong, Web Intelligence, Springer-Verlag, Berlin, 2003.
[15] S.K. Pal, V. Talwar, P. Mitra, Web mining in soft computing framework: relevance, state of the art and future directions, IEEE Transactions on Neural Networks 13 (5) (2002) 1163–1177.
[16] M. Perkowitz, O. Etzioni, Towards adaptive web sites: conceptual framework and case study, Artificial Intelligence 118 (1–2) (2000) 245–275.
[17] T.A. Runkler, J. Bezdek, Web mining with relational clustering, International Journal of Approximate Reasoning 32 (2–3) (2003) 217–236.
[18] J. Srivastava, R. Cooley, M. Deshpande, P. Tan, Web usage mining: discovery and applications of usage patterns from web data, SIGKDD Explorations 1 (2) (2000) 12–23.
[19] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, 1999.
[20] J.D. Velásquez, A study on intelligent web sites: towards the new portals generation, Dissertation for degree of Doctor of Philosophy. University of Tokyo, Tokyo, Japan, 2004.
[21] J.D. Velásquez, P. Estévez, H. Yasuda, T. Aoki, E. Vera, Intelligent web site: understanding the visitor behavior, Lecture Notes in Computer Science 3213 (1) (2004) 140–147.
[22] J.D. Velásquez, R. Weber, H. Yasuda, T. Aoki, A methodology to find web site keywords, in: Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service. Taipei, Taiwan, March 2004, pp. 285–292.
[23] J.D. Velásquez, H. Yasuda, T. Aoki, R. Weber, A generic data mart architecture to support web mining, in: Proceedings of the 4th International Conference on Data Mining. Río de Janeiro, Brazil, December 2003, pp. 389–399.
[24] J.D. Velásquez, H. Yasuda, T. Aoki, R. Weber, A new similarity measure to understand visitor behavior in a web site, IEICE Transactions on Information and Systems, Special Issues in Information Processing Technology for web utilization E87-D (2) (2004) 389–396.
[25] J.D. Velásquez, H. Yasuda, T. Aoki, R. Weber, E. Vera, Using self organizing feature maps to acquire knowledge about visitor behavior in a web site, Lecture Notes in Artificial Intelligence 2773 (1) (2003) 951–958.

Please cite this article as: Juan D. Velásquez, Vasile Palade, A Knowledge Base for the maintenance of knowledge extracted ..., Knowledge-Based Systems (2006), doi:10.1016/j.knosys.2006.05.015