Proceedings of the 5th WSEAS Int. Conf. on DISTANCE LEARNING AND WEB ENGINEERING, Corfu, Greece, August 23-25, 2005 (pp151-157)

Unsupervised Reconstruction Mechanism to Recover the Hypermedia Structure of Instructional Materials on the Web based on the Association Lattice of Keywords

Chang-Kai Hsu, Jyh-Cheng Chang, Maiga Chang[2] and Jia-Sheng Heh
[1]Dept. of Information and Computer Engineering, Chung Yuan Christian University, Taiwan
[2]Program Office of National Science and Technology Program for e-Learning in Taiwan

Abstract:
When a learner is reading an instructional material on the web, sometimes he/she may not understand the meaning of a specific keyword clearly. Therefore, the learner will need more references for that keyword at that moment. However, unfortunately, in the most of time, the learner will not be able to find out. It is because of the instruction designers of learning materials who had never thought that will be a question mark in the learners' mind. Therefore, if the appropriate dependent documents that are associated with the keyword that the learner is looking for could be retrieved automatically and the original document structure could be reconstructed to more suitable for learning and reading, that will be perfect. In this paper, the data mining technique – association rule methodology (ARM) is applying to analyze and using to reconstruct the necessary instructional materials on the web automatically.

Keywords: Association rule, Keywords, Lattice, e-Learning, Instructional materials, Hypermedia, WWW.

1 Introduction

Internet-Based Education is "using the network to transmit, pick and fetch the learning information and contents", including the information scientific, technology, and many kinds of teaching contents. [1][2][5] Rosenberg thoughts that there should be three characteristics when teaching via network: network teaching is networked, can upgrade in time, deposit and withdraw, spread and share the content of courses and information; network teaching uses the standard network technologies, such as TCP/IP, HTML, to teach through the computer environment; and, moreover, the network is not only convey the instructional contents, but also include examination of the learning effects.

By applying data mining techniques suchlike association rules, the partial of instructional materials and/or other structured documents in the hypermedia environment could be seen as transactions in commerce. Moreover, those keywords exist in the materials and the documents could be also used as the frequent (purchasing or buying) items of customers (documents and/or learners). With these assumptions, this paper tries to present an unsupervised reconstruction mechanism in order to recover the hypermedia structure of instructional materials and use the discovered association rules to navigate students in learning.

Section 2 defines the documents' structure and knowledge navigation in the hypermedia environment. In Section 3, ***Association Rules Methodology*** (ARM) and documents' keywords are integrating with the lattice theory in order to analyze the association lattice. Experiment system is implemented for testify the association lattice approach by Section 4. Section 5 makes a simple conclusion and discusses possible future works.

2 Hypermedia structure formulation

As we know, no matter whether a document is, such as a whole book, an article, a paragraph, a section, a chapter, and even one single webpage, which can be seen as a kind of instructional materials. Keywords are some pre-selected terms that can be used to refer to a document and/or an instructional material. In general, most of keywords are nouns.

Keywords are utilized to index and summarize a document's content. For a given set of keywords, not all of these terms are equally useful for representing document's contents. Therefore, when keywords are used in describing a document's contents, each keyword will have various relationships with these contents obviously. These relationships shall be able to express as "association rules" which are generated via some kinds of data mining techniques.

An association rule can be represented by $X \Rightarrow Y$, where *X* means the cause statement and Y means the result statement. With the association rules, users will be able to find some patterns which might hide inside a large mount of data and would be interested by them. For example, a vendor may want to figure out what kinds of item combinations will have good sell. The goal of using such kind of data mining techniques is to automate the process of finding relative patterns and trends[3]. Since the *Apriori* [7] algorithm is the most well-known association rule algorithm and is used in most commercial products, this paper develops the experiment system based on Apriori algorithm.

In general, most of the association mining works have concentrated on the task of discovering the frequent itemsets. There is only very little attentions has paid in extracting rules, and it is so-called the ***rule generation***. Some researches tried to form rules to cover all of data [4][9]. Other works addressed the problems when researchers were trying to figure out the association rules which would be interested [6][8].

First of all, several definitions are needed to describe. A paragraph in a document is a ***transaction***, and each keyword is an ***association property*** which is used in representing the transaction. Let t be the number of keywords and k_i be a generic index. Therefore, the keyword set is $K = \{k_1, k_2, ..., k_t\}$.

Nelson(1995) suggested an idea about associating hypertext in a multiple and flexible way for any hypermedia [10]. Network structure is one possible metaphor to represent the information of nodes and links in a hypermedia environment. No matter a structured document is hypertext or not, which can be transformed into the independent hierarchical (no-flat) structures, is so-called the ***document structure***, $S = \{s_i\}$. The elements of the document structure include chapters, sections, paragraphs, and sentences.

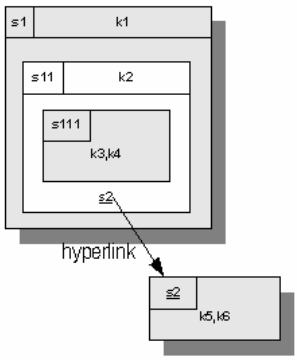

Fig.1. s_1 and s_2 are structured documents.

Example 1. Figure 1 shows an example of hypermedia environment. Here, the document structure s_1 (e.g., article) has a header associated with the keyword k_1 and a paragraph s_{11}. The structure s_{11} also has another header associated with the keyword k_2, one sentence (s_{111}), and one hyperlink to the document s_2. The sentence s_{111} contains two keywords, they are k_3 and k_4. The document s_2 contains another two keywords – k_5 and k_6. □

In the most of e-Learning systems, instructional materials (or courseware) editors often focus on the content design, such as pictures, animation objects, video games, and online-test. However, the instruction designers may not pay attentions in considering the learners' possible traversal paths. Different traversal paths (or learning paths) may let learners feel comfortable or confused, especially when the instruction materials are designed for a hypermedia environment such like WWW.

Example 2. When learner reading a courseware of *Java language*. Taking the Sun's Java online lectures for example (http://java.sun.com/docs/books/tutorial/java/index.html). This courseware was writing about the concepts of object-oriented programming. In this document, there is a statement - "…A software object implements its behavior with *methods*…". When a learner read this statement, he/she may not understand the meaning of the keyword – "methods" clearly. Therefore, the learner will need more references for "methods". However, unfortunately, in the most of time, the learner will not find out. It is because of the designers of instructional materials had never thought that will be a question mark in the learners' mind. Therefore, if the dependent documents that are associated with the keyword – "methods" could be retrieved automatically and the original document structure could be re-constructed to more suitable to learn and read for individual learner just like Fig. 2 shows, that will be perfect. □

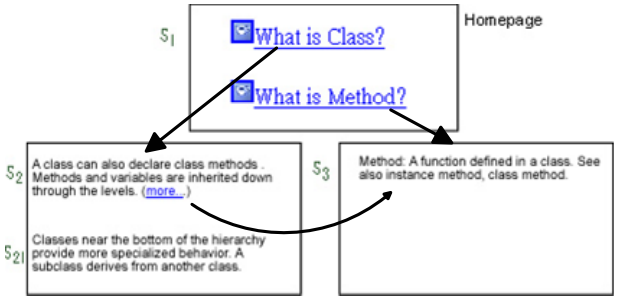

Fig. 2. re-construct the original webpage and provide related webpage's link automatically.

In order to reach the objective that is mentioned above, there are two major problems that should be discussed and analyzed first.
(a). *Association Lattice of Keywords:* given a set of keywords, how to construct an association lattice from these keywords by retrieving the contents of the original hypermedia instructional materials?
(b). *Reconstructing Hypermedia Structure:* after the association lattice of keywords worked out, how to reconstruct two or more hypermedia structured documents?

3. Association Lattice of Keywords

The task of mining associations between keywords can be stated as follows: Let $K = \{k_1, k_2, ..., k_m\}$ be a set of keywords, and let $S = \{S_1, S_{1i}, ...S_{n1}, ...S_{ni}\}$ be a set of structure identifiers for hypermedia documents. The input database in a binary relation $\delta \subseteq K \times S$. If a keyword k_i is found in a structure S_j, we write it as $(k_i, S_j) \in \delta$ or alternately as $k_i \delta S_j$. In typical, a hypermedia instructional material contains lots of document structures, where each document structure contains a set of keywords.

The ***support*** of the keyword set K is denoted as $\partial(K)$. The value of $\partial(K)$ is the number of document structures in which there is a same keyword exists. A keyword set will be ***frequent*** if its support value is higher than the expected threshold, $\partial(K) \geq minsup$, where *minsup* is a user-specified ***minimum support*** threshold.

Example 3. Consider the hypermedia instructional materials shown in Table 1. (Table 1 will be used as a running example in the whole paper). The keywords in the keyword set $K = \{O, C, M, I, V\}$ are "Object", "Class", "Method", "Inheritance", "Variable". Moreover, the document structure set is $S = \{S_1, S_2, S_3, S_4, S_5, S_6\}$. In order to simplify and make it more readable, in the following examples, the $S_2 = \{C\delta 2, M\delta 2, V\delta 2\}$ will be replaced by *CMV* when the document structure contains a keyword set – $\{C, M, V\}$. Similarly, a document structure set $\{1,3,5\}$ will also write as 135 for the same reason. □

S_i	**Keywords in S_i**
1	*OCIV*
2	*CMV*
3	*OCIV*
4	*OCMV*
5	*OCMIV*
6	*CMI*

Table 1. Five keywords in six document structures.

Frequent keyword set *minsup* >= 50%	**Support**
C	100%
V, CV	83%
O, M, I, OC, OV, CM, CI, OCV	67%
OI, MV, IV, OCI, OIV, CMV, CIV, OCIV	50%

Table 2. Frequent keyword sets in the structured documents with *minsup*=50%.

A ***confident association rule*** is denoted as $\{k_1, k_2, ..., k_n\} \xrightarrow{p} k_m$, which means that if all of the keywords $k_1, k_2, ..., k_n$ can be found out in the specific document structure, then there should be a good chance to find the keyword k_m out in the same structure, too. The acceptance ratio (probability) for such kind of association rules is called the ***confidence*** of the rule, and the probability is denoted by p. The probability p can be calculated by $p = \partial(k_1 k_2 ... k_n \cup k_m) / \partial(k_1 k_2 ... k_n)$. In practical, researchers only care those association rules with high confidence, that is $p \geq minconf$.

Example 4. One extracted rule for Java instructional materials on the Sun website is $\{subclass, superclass\} \xrightarrow{0.85} inheritance$. This rule means that the document structure in the instructional material mentions the "subclass" and "superclass" will also mentions the "inheritance". And this rule's confidence will be 0.85 (85%). □

As *Apriori* uses the ***frequent itemset*** to generate rules, a frequent itemset is an itemset whose occurrence ratio is higher than the threshold. An itemset can be seen as a keyword set (K_i) in the instructional materials. Beside the itemset, the ***transaction set*** in *Apriori* can be also represented as the document structure (S_i). By using *Apriori* a association lattice of keywords (based on Example 4) can be built more easily as Fig. 4 shown.

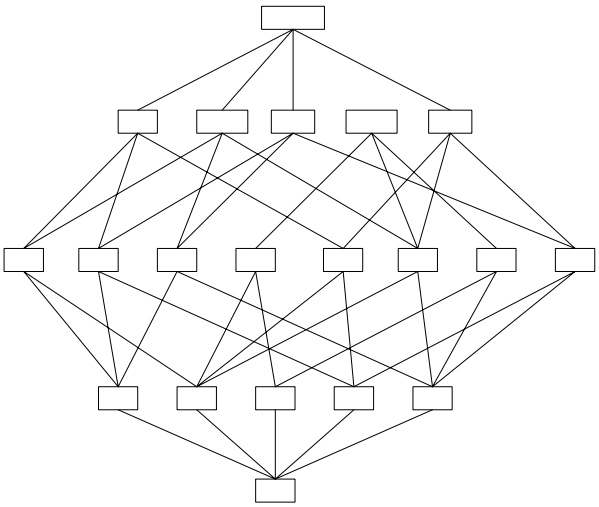

Fig. 4 Association lattice of keywords $\{O, C, M, I, V\}$ based on Example 4.

In this case there are five keywords (items) $\{O, C, M, I, V\}$. The line in the lattice represents the relation between two keyword-sets (itemsets). The frequent itemset property in *Apriori* mentioned that any subset of an itemset must be frequent if the parent itemset is frequent. Figure 5 shows the nonempty subsets of *OCV* are $\{OC, OV, CV, O, C, V\}$. Therefore, according to the frequent itemset property if *OCV* is frequent, then all of the subsets should be frequent, too.

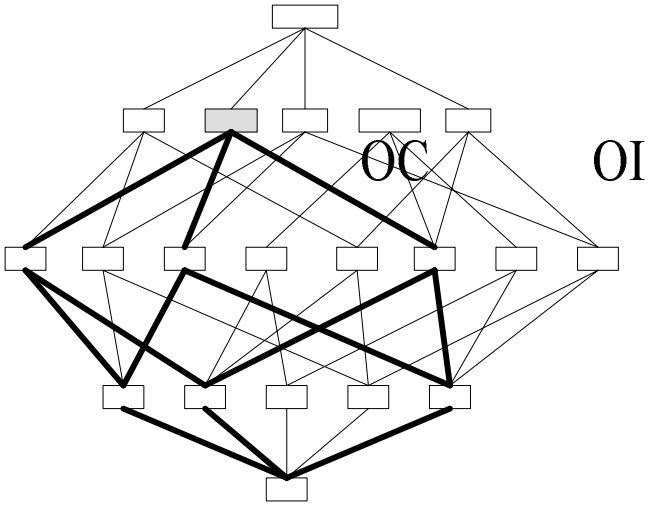

Fig. 5 Subsets of *OCV*.

After an association lattice of keywords was constructed, the next step is generating confident rules. The step can be divided into three stages:

1. To choose the frequent itemset with a minimal support threshold. For example, when the *minsup*=50%:
   (1) *C* with support 100%;
   (2) *V*,*CV* with support 83%;
   (3) *O*, *OC*, *OV*, *OCV* with support 67%.
2. To find out all the possible rules: (according to the example above)
   (1) $O \rightarrow CV$; (4) $OV \rightarrow C$;
   (2) $OC \rightarrow V$; (5) $V \rightarrow OC$;
   (3) $C \rightarrow OV$; (6) $CV \rightarrow O$.
3. To search the confident association rules with a minimal confidence threshold. For example, when the *minconf* =100%:
   (1) $O \xrightarrow{1.0} CV$;
   (2) $OC \xrightarrow{1.0} V$;
   (3) $OV \xrightarrow{1.0} C$.

By using the three stages above, the association lattice of keywords can be retrieved automatically For example, if a learner wants to study about the topics – $\{O, C, V\}$, the relevant association lattice of keywords *OCV* can be discovered as shown in Fig. 6.

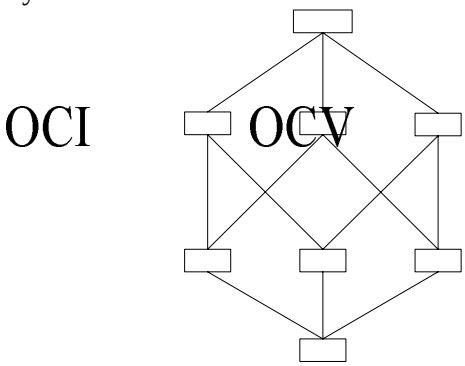

Fig. 6 Association lattice of keywords with a learning topic $\{O,C,V\}$.

4 Experiment Framework

This paper applies the experiment system to the instructional materials of Java language on the Sun's website. There are five keywords ("Object", "Class", "Method", "Inheritance", "Variable") and six document structures. The contents of original document structures are shown in Table 3. After parsing the text of those documents, the keyword set can be derived as mentioned in Table 1.

S_1	Software objects are modeled after real-world objects in that they too have state and behavior. A software object maintains its state in one or more variables. Classes near the bottom of the hierarchy provide more specialized behavior. A *subclass* derives from another class. For example, a subclass cannot access a private member inherited from its superclass

S_2	A class can also declare class methods . Methods and variables are inherited down through the levels. In general, the farther down in the hierarchy a class appears, the more specialized its behavior. In addition to instance variables,

	classes can define class variables .

S_3	Object-oriented programming, Subclasses can also override inherited methods and provide specialized implementations for those methods. You are not limited to just one layer of inheritance. The inheritance tree, or class hierarchy, can be as deep as needed. A class variable contains information that is shared by all instances of the class. In such situations, you can define a class variable that contains the number of gears. All instances share this variable.

S_4	These objects are created when the user launches the application. The application's main method creates an object to represent the entire application, and that object creates others to represent the window, label, and custom component. You can invoke a class method directly from the class, whereas you must invoke instance methods on a particular instance. If one object changes the variable, it changes for all other objects of that type.

S_5	Because the object that represents the spot on the screen is very simple, let's look at its code. The Spot class declares three instance variables: size contains the spot's radius, x contains the spot's current horizontal location, and y contains the spot's current vertical location. It also declares two methods and a constructor — a subroutine used to initialize new objects created from the class. The inheritance tree, or class hierarchy, can be as deep as needed.

S_6	Method: A function defined in a class. See also instance method, class method. Unless specified otherwise, a method is not static. Methods are inherited down through the levels.

Table 3. Contents of hypermedia instructional materials with specific keywords.

The relations of keyword set can be presented by the association lattice as shown in Fig. 7. After applying the minimal support threshold and the minimal confidence threshold to the lattice, some association rules can be retrieved. For example, five most general rules are:

(1) $\{Inheritance, Varible\} \rightarrow Object$;

(2) $Object \rightarrow Variable$; (4) $Inheritance \rightarrow Class$;

(3) $Variable \rightarrow Class$; (5) $Method \rightarrow Class$.

Base on these association rules, when a learner is reading S_2 in which there are "*Inheritance*" and "*Variable*", the system can advice him to read another document in which the "*Object*" is included. For example, S_4 is a document which is talking about the concept "*Object*".

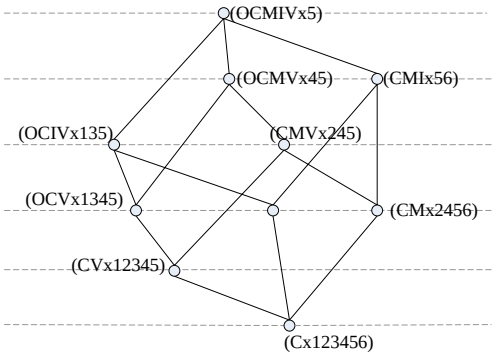

Fig. 7. Association lattice of the instructional materials according to Table 1 and Table 3.

To realize the mechanism proposed in this paper, an internet-based learning system called *Knowledge Enhance Network* (**KEN**) is implemented as shown in Fig. 8. Fig. 9 shows the operational flow of analyzing document structures and keywords, constructing the association lattice of keywords, and reconstructing the hypermedia instructional materials automatically.

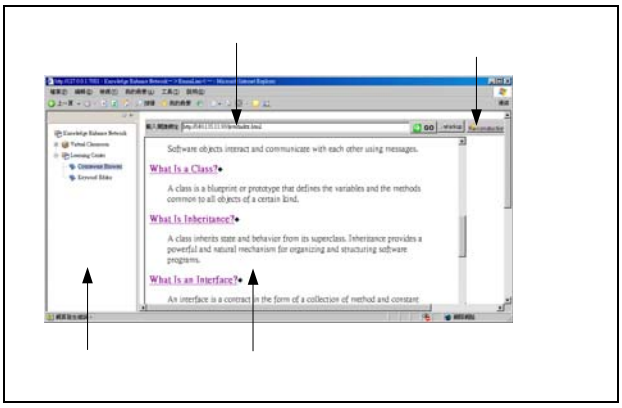

Fig. 8. Shap-shot of **KEN**.

Choosing the topic – "What is Class?", and feed it into **KEN** as shown in Fig. 10(a). By click the "Re-construct" button, a hyperlink will automatically add on the document, see the Fig. 10(b). If a learner wants to see more relevant documents, then he/she can just click the hyperlink and the relevant document that is associated with the source document will be displayed in the reading area, as shown in Fig. 10(c).

5 Conclusions

A web-based instructional material analyzing and reconstructing system is built in this paper according to the unsupervised reconstruction mechanism. By using association lattice of keywords, the necessary links between learning resources can be generated automatically for individual learner. Moreover, the

most appropriate instructional materials (webpages) can also be able to find out for making suggestions to learners. However, the working procedure is still a little of complicated. The experiment framework was really created to prove that the reconstruction mechanism is workable and a reasonable association rules were also discovered. It is worth to note that the experiment system is not only can be used for e-Learning system, but also will be available for any kind of web-based browsing environment. From now on, there are still several works can be done: the association lattice of keywords should be generated in real time; using *minconf* = 100% may lose some information; and, the keyword database may be able to construct automatically or semi-automatically.

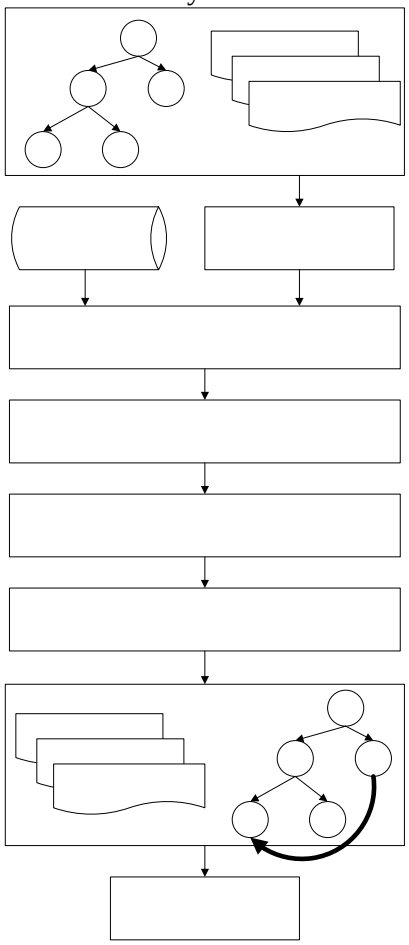

Fig. 9. The operational flow.

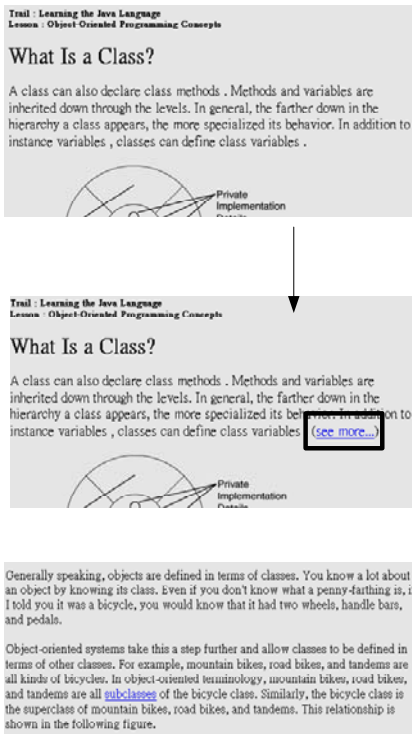

Fig. 10. Example of reconstructing a hypermedia instructional material automatically.

References:

[1] D. Persico, Methodological Constants in Courseware Design, *British Journal of Educational Technology*, Vol. 28, No. 2, pp. 111-123, 1997.

[2] G. A. Novak, Virtual Courseware for Geoscience Education: Virtual Earthquake and Virtual Dating, *Computers & Geosciences*, Vol. 25, No. 4, pp. 475-488, 1999.

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, San Mateo, CA: Morgan Kaufmann Publishers Inc., 2001.

[4] J. L. Guigues and V. Duquenne, Familles Minimalesd'implications Informatives Resultant d'un Tableau de Donnees Binaires, *Math. Sci. hum.*, Vol. 24, No. 95, pp. 5-18, 1986.

[5] J. Phelps and R. Reynolds, Formative Evaluation of a Web-based Course in Meteorology, *Computers & Education*, Vol. 32, pp. 181-193, 1999.

[6] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, Finding Interesting Rules From Large Sets of Discovered Association Rules. *In the proceedings of the 3rd International Conference on Information and Knowledge Management*, Nov. 29-Dec. 2, 1994, Gaithersburg, MD, USA, pp. 401-407, 1994.

[7] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, Fast Discovery of Association Rules, *Advances in Knowledge Discovery and Data Mining*, pp. 307-328, 1996.

[8] R. J. Bayardo and R. Agrawal, Mining the Most Interesting Rules. *In the proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 15-18, 1999, San Diego, California, USA, pp. 145-154, 1999.

[9] R. T. Ng, L. Lakshmanan, J. Han, and A. Pang, Exploratory Mining and Pruning Optimizations of Constrained Association Rules. *In the proceedings of the ACM SIGMOD International Conference on Management of Data*, Jun. 2-4, 1998 , Seattle, Washington, USA, pp. 13-24 1998.

[10] T. Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, New York: HarperCollins Publishers Inc., 1999