Volume 2, No. 8, August 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
REVIEW ARTICLE
Available Online at www.jgrcs.info

STORE-AND-SEARCH: A MODEL FOR KNOWLEDGE DISCOVERY

Dr. S.S. Dhenakaran*[1], S.Yasodha[2]
*[1] Asst. Prof. in Computer Science, Alagappa University, Karaikudi, Tamilnadu, India
[2] Asst. Prof. in Computer Science, Govt. Arts College (W), Pudukkottai, Tamilnadu, India
[2]yasodha_karthik@yahoo.co.in

***Abstract**:* The combination of two powerful technologies - the Semantic Web and Data Mining - will probably bring the internet and even the intranet closer to human reasoning than we ever thought possible. The internet is simply viewed as one huge, distributed database just waiting to be made sense of. Preliminary work in transforming this huge corpus of text, images, sound and video is already available. There is still a long way to go until efficient algorithms for automatic conversion of traditional data into ontologies and concept hierarchies will be found.

In this paper we present two approaches to semantic web mining, each concerning a different aspect –yet focusing on the same basic problem: making sense of already-existing data designed originally only for human readers. The first one is an approach to recurring pattern mining and the second is a *store-and-search* model for knowledge discovery. We present in this paper only a small subset of work undergone in this exciting field of Semantic Web Mining, but we hope that it will provide a glimpse into the realm of possibilities that it opens.

***Keywords**:* Ontology, inference, knowledge discovery, information gathering, information filtering, overload.

INTRODUCTION

Over the last decade, there is an explosive growth in the information available on the Web. Today, web browsers provide easy access to myriad sources of text and multimedia data. More than one billion pages are indexed by search engines, and finding the desired information is not an easy task. One of the fundamental issues regarding the efficiency of information gathering (IG) is “overload” [1]. The problem of information overload occurs when a large number of irrelevant documents may be considered to be relevant. The existing approaches of information retrieval (IR) and information filtering (IF) could be used to solve this problem [2] [6]. The problem, however, is that most approaches of IR and IF cannot explicitly interpret user profiles (e.g., the user feedback, or the user log data).

The human ability for information processing is limited on the one hand, whilst otherwise the amount of available information of the Web increases exponentially, which leads to increasing information saturation [3]. In this context, it becomes more and more important to detect useful patterns in the Web, thus use it as a rich source for data mining [4].

Web intelligence (WI) is a new direction to push technology toward manipulating the meaning of Web data and creating a distributed intelligence that can actually get things done [5].

Traditionally, the knowledge engineers spend much time in the acquisition of knowledge from domain experts that is the “knowledge acquisition bottleneck” problem. Machine learning and evolutionary computing approaches have all been shown to have niches in which they perform well. In most cases, however, these approaches have either not had the impact of the expert systems produced by the usual knowledge engineering methods; or they have required significant domain expertise for the design of the algorithms, training set, etc [9]. Furthermore, the knowledge acquisition bottleneck in Web-based intelligent information systems becomes particularly difficult, because such systems must have a time-consuming and centralized domain knowledge engineering for ubiquitous information.

The semantic Web is a step towards Web intelligence. It is based on languages that make more of the semantic content of the webpage available in machine-readable formats for agent-based computing. One of the components of semantic Web techniques is to use ontology for marking up Web resources and assisting the generation and processing of semantic markup. This need forces the Web users to use more meaningful XML documents and more new semantic markup languages to describe information in ontologies. However, manual ontology construction remains a tedious, cumbersome task that can easily result in a bottleneck for WI.

Web mining is a new technology that has emerged as a popular area in the field of WI. Currently Web mining could be viewed as the use of data mining techniques to automatically retrieve, extract, generalize, and analyze information [25]. It is obvious that data mining techniques ([1] [3] [4]) can be used for Web mining. Web mining, however, is very different from data mining in that the former is based on Web-related data sources, such as semi-structured documents (HTML, or XML), log, services, and user profiles, and the latter is based on more standard databases.

The research area of Semantic Web Mining is aimed at combining two fast developing fields of research: the Semantic Web and Web Mining. Web mining is used to automatically discover and extract information from Web-related data sources such as documents, log, services, and user profiles. Although standard data mining methods may be applied for mining on the Web, many specific algorithms need to be developed and applied for various purposes of Web based information processing in multiple Web resources, effectively and efficiently.

In this paper we present two approaches to semantic web mining: The first one is an approach to recurring pattern mining, the second is a *store-and-search* model for knowledge discovery.

BASIC DEFINITIONS

Ontology

An ontology defines the terms used to describe and represent an area of knowledge, explicitly. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains [W3C, 2001].

In this paper, an ontology will be used to represent the domain information for a certain user or a group of users. As we all know, the term of ontology has been used for several ways. These range from simple taxonomies, to metadata schemes, to logical theories. In this paper, we use the ontology to describe the user conceptual level – the "intelligent" part for the ontology.

An ontology consists of classes, relationships and attributes. The classes in an ontology are general things (in the many domains of interest). Usually the names of classes are nouns. The properties (or attributes) are those the things may have.

Knowledge Discovery

Knowledge discovery also known as Knowledge Discovery in Databases (KDD) is the most well-known branch of data mining, that describes the process of automatically searching large volumes of data for patterns that can be considered knowledge about the data. It is often described as deriving knowledge from the input data. Knowledge discovery is a data-triggered process that presumes that we already have a collection of web data and we want to extract potentially useful knowledge out of it.

This complex topic can be categorized according to 1) what kind of *data* is searched; and 2) in what form is the result of the search represented. Just as many other forms of knowledge discovery, Knowledge discovery developed out of the Data mining domain also creates abstractions of the input data. The knowledge obtained through the process may become additional data that can be used for further usage and discovery.

RECURRING PATTERN MINING

Research in the semantic web mining until now was focused on adapting known KDD techniques to the Semantic web by: mining conventional information source (databases, files etc.) to augment a semantic network or extract information from the semantic network and then mine it for conventional purposes. The new challenge involves mining the semantic network to enhance the semantic network itself.

Examples of Semantic Web applications that are available and store information in large (semantic) networks are: ODESeW, SEAL, OntoWeb and SemanticOrganizer. The big issue revolves around how can recurring semantic structure be useful, once it is identified.

A few suggestions would be:

1. Generating inference rules (enforcing logical consistency in semantic network is non-trivial). Use patterns of recurring structures to generate candidate rules.

***2. Aiding in analysis* -** identifying recurring structures in the semantic network is part of the analysis process (in order to make sense of existing information)

***3. Performing ontology maintenance* -** currently takes a lot of time and is very difficult. Recurring patterns can indicate relevant and/or necessary changes to the ontology.

***4. Abstracting recurring structures* -** abstracting recurring structures in order to reduce the complexity of displaying and/or navigating semantic networks. Semantic templates are used to define recurring semantic structure and consist of abstracted RDF-like triples and recurring semantic structures in the network are matches to the triples.

An example is:

(?x player-of ?y)
(?x scored ?z)
(?z highest ?w)

Matches

("Sachin" player-of "Cricket")
("Sachin" scored "18000 runs")
("18000runs" highest "ODIscore")

Example 1. Example of a semantic template and a corresponding match in the semantic network.

Generating Inference Rules

Inference must be used to add links between semantic network nodes that are relevant, but have not been added by users. Possible sources of inference rules are structure of the ontology and domain knowledge. Inference rules can be regarded as composed of semantic templates with antecedent and consequent sections - each consisting of a semantic template. Even unsophisticated techniques that generate a large number of candidates for antecedent and consequent sections can help in identifying rules, since it is easier than doing it manually.

Example:

(?sample gathered-during ?experiment)
(?experiment conducted-at ?site)
->
(?sample collected-from ?site)

Example 2. Example of an inference rule from a biology domain (rules used to enforce semantic consistency).

Aiding in Analysis

Recurring patterns can be used to identify interesting facts about the domain, as this example from a biology related field shows:

(?culture salinity "high")
(?culture pH-level "9.0")
->
(?culture exhibits "speckling")

Example 3. Example of an unexpected rule that reveals a previously unknown correlation.

Statistical analysis on recurring patterns can also generate interesting facts. Let's take some example data about 1000 plane accidents. How can this be useful? Say the Plane manufacturing company expected only 4 accidents to involve jackscrews, but the rule templates return a lot more; this must be an interesting fact. Examples show templates for the above example.

(?accident involves ?plane)
(?plane model "MD-80")
Example 4. A semantic template that has 40 matches.
(?accident involves ?plane)
(?accident concerns "jackscrew-failure")
Example 5. A semantic template that has 100 matches.
(?accident involves ?plane)
(?plane model "MD-80")
(?accident concerns "jackscrew-failure")
Example 6. A semantic template that has 16 matches, indicating a correlation between jackscrew failures and MD-80 accidents.

Performing Ontology Maintenance

Ontologies require maintenance as they evolve. Identifying recurring structures helps in revising ontology concepts. For instance, in a financial portal the concept of *tax* appears, but in real life taxes are often of different types and for financial experts it is difficult to work only with one big concept. So they decide to split this concept into more useful categories, such as: *income tax*, *luxury tax*, *import tax*, *value added tax* etc. This results in more meaningful modeling.
(?tax created-by ?regulating-institution)
(?tax type-of-income ?income-category)
Example 7. An income tax template
(?tax item ?luxury-item)
(?tax area-of-application ?taxable-goods)
Example 8. A luxury tax template
(?tax goods-category ?categories)
(?tax importer (?person-name OR ?company-name))
Example 9. An import tax template

Abstracting Recurring Structures

Semantic networks can become very complex in structure, thus having some disadvantages in visualization – it is difficult to display the entire network. It is also infeasible to just display a few nodes around a semantic node (which would mean insufficient context) because a user would be unable to make sense of it. Displaying the whole graph would bring again the problem of having too many nodes and edges to visualize. The solution would be to use recurring patterns (templates) to summarize similar nodes and edges. So, in order to reduce the complexity of displaying and/or navigating semantic networks, we can abstract recurring structures. Semantic templates are used to define recurring semantic structure and consist of abstracted RDF-like triples.
Example - biological domain where scientists perform experiments and collect measurements.
To abstract specific experiments and measurements we can use the following template.
(?experiment produces ?measurement)
(?measurement collected-on ?date)
(?measurement measures ?sample)
Example 10. An abstracted template.

KNOWLEDGE DISCOVERY

To gather relevant data from the Web or databases, we assume that the system can obtain users' log data or feedback. We also assume that the systems can extract some significant attributes from the data. According to these assumptions, we can use a list of facts to describe what we have obtained from the users.

An example of a list of facts is shown in Table 1. In Table 1, four facts concerning 100 posted job details are obtained from a user's feedback, where, *N* denotes the number of posted jobs in one month, which have the same job titles and similar job descriptions. The job titles are determined by the user according to the job descriptions

The user profiles are stored as ontology in RDF templates, which are then searched for knowledge discovery. The process of knowledge representation as ontology and knowledge discovery is implemented by the proposed model called *store-and-search.*

Table 1: The list of facts

Fact	Job Title	Posted	Location	Pay	Description	*N*
f1	Programmer	DEC-99	Sydney	80K	Java Programmer.	35
f2	Analyst	DEC-99	Sydney	90K	Analyst with skills in C++, UNIX and SQL	22
f3	Analyst	DEC-99	Melbourne	90K	Analyst with experience in Cobol and SQL /DB2.	18
f4	Programmer	MAY-99	Melbourne	80K	Programmer with expertise in ASP, SQL server and Pearl	25

Methodology

Proposed Method

To store the facts, the Semantic Web contains RDF. RDF stores information in the form of XML. In this paper, a new algorithm called *store-and-search* has been proposed for knowledge representation and knowledge discovery which works in two phases:

1. The first phase is the knowledge representation phase or store phase. The store phase stores facts as ontology in the form of RDF templates.
2. The second phase is the knowledge discovery phase or search phase. The search phase performs knowledge discovery by attribute matching.

Semantic web is useful for knowledge discovery by itself without human interaction. So during the first phase, it is necessary to maintain all the details that are required by the computer to satisfy the user's requirements. The server analyzes these facts to identify some significant attributes. These details including the user's login particulars, the facts and the significant attributes are stored in the RDF.

During the second phase when the user enters into the web, the server accepts the user's request and extracts the values of significant attributes. These values of attributes are used as the key for matching of facts. Knowledge discovery is thus done in the second phase.

The store phase in the RDF file is as follows:

```
<?XML Version=”1.0”?>
<!—Store phase-->
<login>
<!-Store Login particulars of the User>
<entry>
<UserID> [User’s ID]</UserID>
<Password>[PASSWORD]</Password>
</entry>
</login >
<Fact table>
<!—Store fact table entries-->
<entry>
<Fact>[FACT]</ Fact >
<Jobtitle>[ JOBTITLE]</ Jobtitle >
<Posted>[ POSTED-DATE]</ Posted >
<Location>[ LOCATION]<./ Location >
<Pay >[ PAY] </ Pay >
<Description>[DESCRIPTION] <./ Description>
<N>[NO. POSTED]</N>
</entry>
</Fact table>
```

The search phase in the RDF file is as follows:

```
<?XML Version=”1.0”?>
<!—Accept User’s request as search keywords-->
<entry>
<Keyword><USER’S REQUEST></ Keyword >
<!—If User’s request matches with the values of stored attributes then extract facts-->
<Output>
<Fact>[FACT]</ Fact >
<Jobtitle>[ JOBTITLE]</ Jobtitle >
<Posted>[ POSTED-DATE]</ Posted >
<Location>[ LOCATION]<./ Location >
<Pay >[ PAY] </ Pay >
<Description>[DESCRIPTION] <./ Description>
<N>[NO. POSTED]</N>
</Output>
</entry>
```

The proposed method is based on the following algorithm which works based on the given conditions:

The store-and-search Algorithm

```
First phase: The store phase
Begin
The Server accepts the user’s facts
Analyze the facts to identify significant attributes
Store the facts and login particulars in RDF XML File.
End
Second phase : The search phase
Begin
The Server Accepts the user’s request
Check the facts stored in store phase
If the user’s request matches the values of significant attributes then
Extract relevant facts
Else
Display “No match found”
End If
End
```

Experimental results

In a large application, if the user wants to retrieve relevant information for an application, it would become intangible when the user sends their request through the Keywords. So in this paper, a Semantic Web Mining model is used to provide the necessary service to the user and fulfill their requirements. It can be applied in many fields such as Web Store, Web Shopping, E-Commerce, E-Jobs, Online Transaction and so on. In this paper, an E-Jobs application is considered.

Consider a large Website for online jobs, which can be used by several users for selecting their profession. In such case, it is the necessary task to store the information about the arrival and departure of the user and also the details like a history or log maintenance in the web through the RDF. Using the RDF file, the web can automatically analyze the user’s request and provide the relevant facts based on keyword matching.

Thus the knowledge discovery model is successfully implemented.

CONCLUSION

The purpose of this paper was not to give an extensive coverage of semantic web mining, but rather to give a general overview of the possibilities that this area opens to future research and applications. We have shown how to mine recurring patterns and represent knowledge as ontology in the form of RDF templates and use this ontology for knowledge discovery. In this paper we presented a model to automatically discover knowledge for a particular user or a group of users. One of the main contributions of this paper is that we use ontology to represent user profiles and this knowledge representation is used by the *store-and-search* model to discover knowledge without human intervention.

REFERENCES

[1]. N. R. Jennings, K. Sycara and M. Wooldridge, A Roadmap of agent research and development, Autonomous Agents and Multi-Agent Systems, 1998, 1(1): 7-38.

[2]. R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999.

[3]. M.-S. Chen, J. Han, and P. S. Yu, Data mining: an overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6):866-883.

[4]. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R.Uthrusamy, eds., Advances in knowledge discovery and data mining, Menlo Park, California : AAAI Press/ The MIT Press, 1996.

[5]. E. A. Feigenbaum, How the “what” becomes “how”, Communications of the ACM, 1996, 39(5): 97-104.

[6]. D. A. Grossman and O. Frieder, Information retrieval algorithms and heuristics, Kluwer Academic Publishers, Boston, 1998.

[7]. J. W. Guan and D. A. Bell, Evidence theory and its applications, Volume 1, Studies in Computer Science and Artificial Intelligence 7, Elsevier, North-Holland, 1991.

[8]. J.W. Guan and D. A. Bell, A generalization of the Dempster-Shafer theory, Proceedings of IJCAI, 1993, 592-597.

[9]. J. Hendler and E. A. Feigenbaum, Knowledge is power: the semantic Web vision, in N. Zhong et al., eds., Web Intelligence: research and development, LNAI 2198, Springer-Verlag, 2001.

[10]. D. A. Hull, and S. Roberston, The TREC-8 filtering track final report, in: TREC-8, 1999.

[11]. R. Janicki, Towards a mereological system for direct products and relations, 3rdInternational Conference on Rough Sets and Current Trends in Computing, Malvern, PA, USA, 2002, 113-122.

[12]. R. Agrawal, T. Imielinski and A. Swami, Database mining: a performance perspective, IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6):914-925.

[13]. R. Kruse, E. Schwecke and J. Heinsoln, Uncertainty and vagueness in knowledge based systems (Numerical Methods), Springer-Verlag, New York, 1991.

[14]. V. Lesser et al., BIG: an agent for resource-bounded information gathering and decision making, Artificial Intelligence, 2000, 118: 197-244.

[15]. V. Lesser and S. Zilberstein, Intelligent information gathering for decision models, Computer Science Technical Report TR-96-35, University of Massachusetts at Amherst, 1996.

[16]. Y. Li, Information fusion for intelligent agent-based information gathering, in N. Zhong et al., (eds.) Web Intelligence: research and development, LNAI 2198, Springer-Verlag, 2001, 433-437.

[17]. Y. Li and Y.Y. Yao, User profile model: a view from artificial intelligence, 3rdInternational Conference on Rough Sets and Current Trends in Computing, Malvern, PA, USA, 2002, 493-496.

[18]. Y. Li and C. Zhnag, A method for combining interval structures, 7th International Conference on Intelligent Systems, Melun, France, 1998, 9-13.

[19]. Y. Li, C. Zhang, and J. R. Swan, A multiple decomposed approach for relevant functions information retrieval, Database, Web and Cooperative Systems, edited by George E. Lasker and Yanchun Zhang, The International Institute for Advanced Studies in Systems Research and Cybernetics publisher, 1, Baden-Baden, Germany, 1999, 97-102.

[20]. Y. Li, C. Zhang, and J. R. Swan, An information filtering model on the Web and its application in JobAgent, Knowledge-based Systems, 2000, 13(5): 285-296.

[21]. T. Y. Lin, Database mining on derived attributes, 3rdInternational Conference on Rough Sets and Current Trends in Computing, Malvern, PA, USA, 2002, 14-32.

[22]. D. Liu and Y. Li, The interpretation of generalized evidence theory, Chinese Journal of Computers, 1997, Vol 20 No 2, pp. 158-164.

[23]. K. C. Lee, J. S. Kim, N. H. Chung and S. J. Kwon, Fussy cognitive map approach to web-mining Knowledge-Based Systems, 2002, 22:197-211.

[24]. J. Mostafa, W. Lam and M. Palakal, A multilevel approach to intelligent information filtering: model, system, and evaluation, ACM Transactions on Information Systems, 1997, 15(4): 368-399.

[25]. S. K. Pal and V. Talwar, Web mining in soft computing framework: relevance, state of the art and future directions, IEEE Transactions on Neural Networks, 2002, 13(5): 1163-1177.