Retrieving Rules Using Composetoonto Based On Stemming Algorithm From Similar Web Sites

B. Priyadharshini[1], A. Mohana[2], R. Radhika[3]

UG Scholar, Dept of UG studies in Engineering, Christ college of Engineering and technology, Puducherry, India[1]
UG Scholar, Dept of UG studies in Engineering, Christ college of Engineering and technology, Puducherry, India[2]
UG Scholar, Dept of UG studies in Engineering, Christ college of Engineering and technology, Puducherry, India[3]

ABSTRACT

Ontology is an explicit specification of conceptualization. Rule acquisition is also an important issue. We expect that it will be easier to acquire rules from a site by using similar rules of other sites in the same domain rather than starting from scratch. We proposed an automatic rule acquisition procedure using rule ontology ComposeToOnto, which represents information about the rule components and their structures. And the Stemming algorithm has been used to search for the exact rules that match to the relevant web sites.

***Index Terms*— Rule acquisition, Rule ontology, ComposeToOnto, Best-first search, Stemming Algorithm.**

1. INTRODUCTION

Web 1.0 concentrated on presenting, not creating so that user-generated content was not available. Web 1.0 [15] was about reading, Web 2.0 [14] is about writing, Web 1.0 was about client-server, Web 2.0 is about peer to peer. Web 1.0 was about HTML, Web 2.0 is about XML, Web 1.0 was about home pages, Web 2.0 is about blogs, Web 1.0 was about services sold over the web, and Web 2.0 is about web services. **The web** is continuously evolving toward web3.0 after going through web2.0.**Web2.0** allows users to interact and collaborate with each other in a social media as creators of user –generated content in a virtual community. To allow users to continue to interact with the page, communications such as data requests going to the server are separated from data coming back to the page (asynchronously). Otherwise, the user would have to routinely wait for the data to come back before they can do anything else on that page, just as a user has to wait for a page to complete the reload. On the server side, Web 2.0 uses many of the same technologies as Web 1.0. Languages such as PHP, Ruby, Perl, Python, as well as JSP, and ASP.NET, are used by developers to output data dynamically using information from files and databases. What has begun to change in Web 2.0 is the way this data is formatted. In the early days of the Internet, there was little need for different websites to communicate with each other and share data. In the new "participatory web", however, sharing data between sites has become an essential capability. To share its data with other sites, a website must be able to generate output in machine-readable formats such as XML and JSON. For example: blogs, video sharing sites, hosted services, web application, etc. Web 3.0 [3] is an extension of web2.0 which is also said to be semantic web where the information's are given in

well-defined meaning better enabling computers and people to do work in co-operation. **Web 3.0** -- will make tasks like your search for movies and food faster and easier. Instead of multiple searches, you might type a complex sentence or two in your Web 3.0 browser, and the Web will do the rest. In our example, you could type "I want to see a funny movie and then eat at a good Mexican restaurant. What are my options?" The Web 3.0 browser will analyze your response, search the Internet for all possible answers, and then organize the results for you. There is several of web ontology like online shopping, online reservation tickets, online banking, research purpose, medical field.

Rule acquisition [11] is an important concept in ontology .It means acquiring rules from similar web sites of same domains. Before rule acquisition machine learning research based on pattern classification and learning by examples was used .But those concepts are different from rule acquisition because machine learning research based on pattern classification and learning by examples provides structured data whereas rule acquisition provides unstructured data.

This paper is organized as follows: in section 2, contains related works including ontology, rule acquisition and best-first search. In section 3, the issues of our approach. In section 4, we proposed a detailed procedure of rule acquisition through ComposeToOnto [7]. In section 5, we have proposed a new algorithm for our work "Stemming Algorithm" that gives an exact result for the needed concepts. In section 6, we have also given the future enhancement of our work to be accomplished. At last, in section 7 presents our conclusions.

2. RELATED WORKS

Ontology learning refers to extracting conceptual knowledge from several sources such as from and building ontology from scratch, enriching, or adapting an existing ontology. Since our part is to provide result in form of rules it is used to identifies classes and instances for the user.

The authors called P. Buitelaar, D. Olejnik, and M. Sintek in their paper A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis [3] which was published in the year 2004 says that Onto Learn is a tool used for an automatic analysis that enables the ontology engineer to bootstrap a domain-specific ontology from document collection. The advantage is it provides precondition language which is used to check some condition and the disadvantage is using OntoLearn [1] tool building ontology for huge amount of data is difficult and time consuming.

The authors called P. Cimiano and J. Volker in their paper Text2onto-a Framework for Ontology-Learning and Data-Driven Change Discovery [6] which is published in the year 2005 says that Text2Onto is a tool used in order to Support the user in constructing ontology's from a huge amount of data given set of (textual) data. The advantages are using this tool building ontology for huge amount of data can be done easily.

The authors called Y. Xu, J. Liu, and D. Ruan in their paper Rule Acquisition and Adjustment Based on Set-Valued Mapping [8] which was published in the year 2003 says that a set valued concept has been introduced to queue problems for objects in presence of multiple attributes. The advantage is Rules are easily acquired using this concept .The disadvantage is time consumption and a complex process.

The authors called M.Y. Dahab, H.A. Hassan, and A. Rafea, in their paper called TextOnto Ex: Automatic Ontology Construction from natural English Text [9] which was published in the year2008 says that TextOntoEx constructs ontology from natural domain text using semantic pattern-based approach.The advantage is support construction of domain relations, non-taxonomic conceptual relationships e.g., causes, caused by, treat, treated by, has-member, contain, material-of, operated-by, controls, etc.

The previously used method is TextToOnto [6] to generate ontology. He said that the TEXT-TO-ONTO Ontology Learning Environment, which is based on a general architecture for discovering conceptual structures and engineering ontologies from text. Our Ontology Learning Environment supports as well the acquisition of conceptual structures as mapping linguistic resources to the acquired structures.

2.1 Why develop ONTOLOGY?

In recent years the development of ontologies—explicit formal specifications of the terms in the domain and relations among them (Gruber 1993)—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become common on the World-Wide Web. The ontologies on the Web range from large taxonomies categorizing Web sites (such as on Yahoo!) to categorizations of products for sale and their features (such as on Amazon.com). The WWW Consortium (W3C) [16] is developing the Resource Description Framework (Brickley and Guha 1999), a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. The Defense Advanced Research Projects Agency (DARPA), in conjunction with the W3C, is developing DARPA Agent Markup Language (DAML) [12] by extending RDF with more expressive constructs aimed at facilitating agent interaction on the Web (Hendler and McGuinness 2000). Many disciplines now develop standardized ontologies that domain experts can use to share and annotate information in their fields.

Some of the reasons for developing ontology:

1. To share common understanding of the structure of information among people or software agents
2. To enable reuse of domain knowledge
3. To make domain assumptions explicit
4. To separate domain knowledge from the operational knowledge
5. To analyze domain knowledge

2.2 Best-First Search

Best-first search [15] is a search algorithm which explores a graph by expanding the most promising node chosen according to a specified rule. Best-first search in its most general form is a simple heuristic search algorithm. "Heuristic" here refers to a general problem-solving rule or set of rules that do not guarantee the best solution or even any solution, but serves as a useful guide for problem-solving. Best-first search is a graph-based search algorithm (Dechter and Pearl, 1985), meaning that the search space can be represented as a series of nodes connected by paths. Some authors have used "best-first search" to refer specifically to a search with a heuristic that attempts to predict how close the end of a path is to a solution, so that paths which are judged to be closer to a solution are extended first. Best-first search in its most basic form consists of the following algorithm (adapted from Pearl, 1984):

The first step is to define the OPEN list with a single node, the starting node. The second step is to check whether or not OPEN is empty. If it is empty, then the algorithm returns failure and exits. The third step is to remove the node with the best score, *n*, from OPEN and place it in CLOSED. The fourth step "expands" the node *n*, where expansion is the identification of successor nodes of *n*. The fifth step then checks each of the successor nodes to see whether or not one of them is the goal node. If any successor is the goal node, the algorithm returns success and the solution, which consists of a path traced backwards from the goal to the start node. Otherwise, the algorithm proceeds to the sixth step. For every successor node, the algorithm applies the evaluation function, *f*, to it, and then checks to see if the node has been in either OPEN or CLOSED. If the node has not been in either, it gets added to OPEN. Finally, the seventh step establishes a looping structure by sending the algorithm back to the second step. This loop will only be broken if the algorithm returns success in step five or failure in step two.

The algorithm is represented here in pseudo-code:

1. Define a list, OPEN, consisting solely of a single node, the start node, *s*.
2. IF the list is empty, return failure.
3. Remove from the list the node *n* with the best score (the node where *f* is the minimum), and move it to a list, CLOSED.
4. Expand node *n*.
5. IF any successor to *n* is the goal node, return success and the solution (by tracing the path from the goal node to *s*).
6. FOR each successor node:
7. Apply the evaluation function, *f*, to the node.
8. IF the node has not been in either list, add it to OPEN.
9. Looping structure by sending the algorithm back to the second step.

3. EXISTING SYSTEM

In existing work, to retrieve the information they have used rule acquisition procedure. In that there is a step called rule identification. Rule identification [5] identifies the rule components such as variables and values from similar web sites. For example, consider a web site www.tatamotors.com, from the ontology it can be easily recognized that delivery of the vehicle and insurance of the vehicle of the web page act as variables and in that scooters, bikes, and cars acts as values.

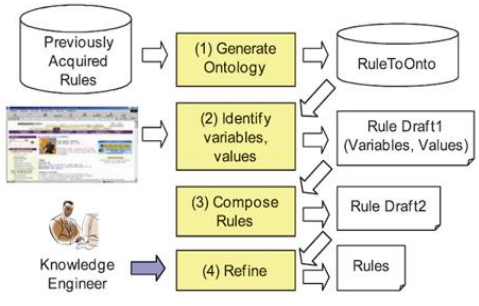

Fig.1 . Overall rule acquisition procedure using *RuleToOnto*.

Diagrammatic description:

The diagram describes that the information about the previously acquired rules has been saved in the form of database. From the stored information of the database, it is used to generate rules from the similar web sites. The rules are acquired using the procedure called rule acquisition that uses a method RuleToOnto. It consists of 2 steps namely,

1. Rule component identification
2. Rule composition

The Rule component identification [5] is used to identify components such as variables and values from the relevant web sites. The identified rule components have been saved in the form of Rule Draft1. The Rule composition is used to compose the components such as variables and values in the form of rules. This produces the result and that is saved as Rule Draft2. Finally, by consulting the Knowledge Engineer i.e. one who is expert in the concepts involved in the Ontology will checks the rules and modifies or adds connectives from the saved Draft and values. The following rule is an example of the refined rule:

The Knowledge Engineer changed the operator of delivery of vehicles from "=" to "<=" and added the value one week. At last it gives the final refined rules to the user from the similar web sites. With the help of the refined rules the user can create their own web sites.

3.1 Problem Definition

The problem that has been identified in the existing method is that the Knowledge Engineer plays an important role to create a website. A knowledge engineer could designate a target range for just one rule in the rule identification step. If there are ten rules in a Web page, the knowledge engineer should divide the area into ten ranges and repeat the rule selection step ten times. That is, there was no rule composition concept in this study. So, the burden on the knowledge engineer is more. This could be a limitation because the results depend on the Knowledge Engineer.

4. PROPOSED WORK

The limitation of the previous work has been overcome by acquiring rules using rule acquisition automatically from the similar web sites. In our paper, we propose a rule acquisition procedure that automatically acquires rules from similar sites by using the rule ontology ComposeToOnto. We propose two main steps for rule acquisition, which consists of rule component identification [5] and rule composition with the identified rule components. In other words, we identify rule components such as variables and values in Web pages by using RuleToOnto [8] in the first step, and we combine the variables to compose rules in the second step.

4.1 Advantages

1. The purpose of using ontology in our approach is to automate the rule acquisition procedure.
2. The starting point of our approach is that it will be helpful for acquiring rules from a site, if we have similar rules acquired from other similar sites of the same domain.
3. It has the advantage that it is structured information and is much smaller than rule bases, so that it is easy to reuse, share, and accumulate.

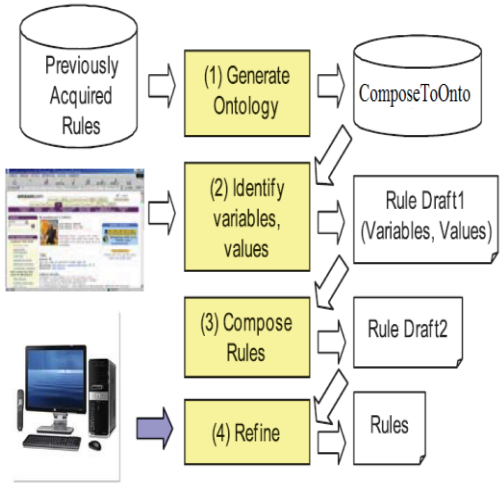

Fig.2. Block diagram for proposed work

4.2 Modules

There are four modules in our proposed work. They are:

1. Rule Ontology Generation
2. Variables and Values Identification
3. Automatic Rule Composing
4. Rule Refinement

1. Rule Ontology Generation:

RuleToOnto is domain specific knowledge that provides information about rule components and structures. The RuleToOnto schema has three object properties Has Value, IF and THEN, and three classes, Variable, Value, and Rule.

2. Variables and Values Identification:

The goal of rule component identification is to elicit variables and values by comparing parsed words of the given text with the variables and values of ComposeToOnto.

3. Rule Composing:

Rules are automatically composed by combining the identified variables and values. There are several possible variable instances for one variable on a Web page. The first step of rule composition is the preparation step, where we find appropriate rules from ComposeToOnto. This is done by comparing the identified variable instances with the variables of the rules in ComposeToOnto.

4. Rule Refinement:

In this module the automatic rule acquisition checks the rules and modifies/adds connectives and values. The following rule is an example of the refined rule. The knowledge engineer changed the operator of days_of_shipment from "=" to "<="and added the value full by referencing the ontology and the target Web page.

5. STEMMING ALGORITHM

1. Put the start node, s on a list called OPEN of unexpanded nodes
2. If OPEN =0 then
3. Exit – no solution exists.
4. Remove a node n from OPEN at which $f=\max(f(n)=gp_{s\text{-}n}(n)+h(n)|n\varepsilon P_{s\text{-}n})$
5. If n is a goal node then
6. Exit with solution
7. Expand node n generating all its successors with pointers back to n
8. For all successor n' of n do
9. Calculate f(n')
10. If n' ~~ε~~ OPEN And n' ~~ε~~ Closed then
11. Add n' to OPEN
12. Assign the newly computed f(n') to node n'
13. Else
14. If new f(n') value is smaller than the previous value, then update with the new value (and predecessor)

15. If n' was in closed, move it back to OPEN

16. Go to (2)

5.1 Algorithm description

The starting node is considered as S and declares that node to be OPEN. If it is OPEN and equal to zero then that shows there is no solution and gets exited from the traversal. And also remove the node that has no solution in it. To compute the minimal optimum solution there is a formula to compute:

$$F=\max (f(n) = g(n) + h(n))$$

If the node n is a goal node then it means there exists a solution so we can traverse through that path. Expanding the OPEN node, for all successor of n then the formula is calculated to find the exact solution that is relevant to our web sites. So now to consider all the nodes to be OPEN assign the computed f(n) and compare it with each goal node. If that goal node is optimum then that is considered as exact solution. So now the traversal can be made on all nodes, then those nodes that has not been visited is considered as OPEN and the steps is as same as from the second step.

6. CONCLUSION

The knowledge engineer's role in rule acquisition is still important, because not all contents of Web pages include rules. Therefore, the knowledge engineer should select the proper parts of Web pages that are expected to have rules in their contents before the rule identification step. Moreover, choosing the Web pages also depends on the Knowledge Engineer's decision. This preliminary work largely affects the performances. If a knowledge engineer can select the exact part of the Web page that contains rules, the performance will be enhanced compared to the case of selecting the whole page. The rule composition retrieves a combination of similar rules for a given range and automatically assigns variable instances to the rules. Thus, by using Rule-based ontology [6] concept we overcome the problems by creating a new website and comparing its performance with the previous work.

We use Rule component identification and Rule composition in Rule acquisition procedure. And also a new algorithm has been proposed to acquire rules automatically from similar web sites using ComposeToOnto [7] method. This algorithm is named as Stemming algorithm which is used to provide the exact solutions to the relevant web sites or domains that the user needs.

7. FUTURE ENHANCEMENT

One limitation of our approach is that the experiment results do not show that the performance of our approach is better than others, because there is no other rule acquisition study that we can directly compare our results with. There are several challenging research issues that must be addressed in order to meet the ultimate goal of our research. First, we are planning to develop a screening method to select exact parts that contain rules from Web pages. Second, we need to extend our research into various domains, because the performance may depend upon the nature of the Web pages in each domain.

8. REFERENCES

[1] P. Velardi, R. Navigle, A. Cucchiarelli, and F. Neri, "Evaluation of OntoLearn, a Methodology for Automatic Learning of Domain Ontologies," Ontology Learning from Text: Methods, Applications and Evaluation, P. Buitelaar, P. Cimiano and B. Magnini, eds. IOS Press, p. 123, 2005.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific Am. Magazine, 2001.

[3] P. Buitelaar, D. Olejnik, and M. Sintek, "A Prote´ge´ Plug-in for Ontology Extraction from Text Based on Linguistic Analysis,"
Proc. First European Semantic Web Symp. (ESWS), 2004.

[4] S. Chae, "Ontology-Based Intelligent Rule Component Extraction," Master Thesis, 2006.

[5] J. Kang and J.K. Lee, "Rule Identification from Web Pages by the XRML Approach," Decision Support Systems, vol. 41, no. 1, pp. 205-227, 2005.

[6] P. Cimiano and J. Volker, "Text2onto-a Framework for Ontology Learning and Data-Driven Change Discovery," Proc. 10th Int'l
Conf. Applications of Natural Language to Information Systems
(NLDB), pp. 227-238, 2005.

[7] C. Golbreich, "Combining Rule and Ontology Reasoners for the
SemanticWeb," Proc. RuleML, pp. 6-22, 2004.

[8] Y. Xu, J. Liu, and D. Ruan, "Rule Acquisition and Adjustment
Based on Set-Valued Mapping," Information Sciences, vol. 157,
no. 1/2, pp. 167-198, 2003.

[9] M.Y. Dahab, H.A. Hassan, and A. Rafea, "TextOntoEx: Automatic Ontology Construction from Natural English Text," Expert Systems with Applications, vol. 34, no. 2, pp. 1474-1480, 2008.

[10] J.C. Beck and M. Fox, "A Generic Framework for Constraint
Directed Search and Scheduling," AI Magazine, vol. 19, no. 4,
pp. 101-130, 1998.

[11] S. Park, J. Kang, and W. Kim, "Rule Acquisition Using Ontology Based on Graph Search," J. Korean Intelligent Information System, vol. 12, no. pp. 95-110, 2006.

[12] I. Horrocks, "DAML+OIL: A Description Logic for the Semantic Web," IEEE Data Eng., vol. 25, no. 1, pp. 4-9, Mar. 2002.

[13] R. Dechter and J. Pearl, "Generalized Best-First Search Strategies and the Optimality of A*," J. the Assoc. for Computing Machinery, vol. 32, no. 3, pp. 505-536, 1985.

[14] T. O'Reilly, "What is Web 2.0," http://oreilly.com/web2/archive/what-is-web-20.html, 2005.

[15] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Member Submission, http://www.w3.org/Submission/2004/SUBMSWRL20040521/,2004.

[16] M.K. Smith, C. Welty, and D. McGuinness, "OWL Web Ontology Language Guide," http://www.w3c.org/TR/owl-guide/, 2004.