Automatic Semantic Rule Extraction in Similar Web Sites using Memetic Algorithm

M. Yogeshwari\textsuperscript{1}, S. Rejeeramani\textsuperscript{2}, G. Senthil Kumar\textsuperscript{3}
\textsuperscript{1}PG Scholar, \textsuperscript{2}Assistant Professor, SVS College of Engineering, Coimbatore
\textsuperscript{3}Assistant Professor, Anna University, Chennai

Abstract— Acquiring knowledge is based on the type of ontology in web mining, and using ontology as a knowledge schema in the knowledge acquisition is more common than rule acquisition. As ontology acquisition the Rule acquisition is an important, even supposing rule acquisition is still a slow down process in the deployment of rule-based systems. Nevertheless, from time to time rules have already been implied in Web pages, in addition to it is possible to acquire them from Web pages in the same manner as ontology knowledge. RuleToOnto process is extracts rules from various domains and web pages not particular domain. In offered system an optimized rule of acquisition method projected with our method through the concept of selecting exact parts that contain rules from Web pages by Genetic Algorithm. But, several proper undesirability results of have been obtained for related problems giving some hope to a more formal handling. To resolve this problem in this work proposed a memetic algorithm as an evolutionary algorithm that incorporates knowledge about the problem domain being resolved. In memetic algorithms using approximate evaluation is more robust than in single-solution methods and the accuracy of the rule acquisition is improved rapidly. As well it provides a relevance score for a web page into annotated result set on user query, and the page annotation, and also decreases the time complexity.

Keywords—RuleToOnto, Rule acquisition, Genetic Algorithm, Memetic Algorithm, Breath first Search, Rule ontology

I. INTRODUCTION

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Companies with a strong consumer focus - retail, financial, communication, and marketing organizations, primarily use data mining today.

It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. The Semantic Web is an evolving development of the World Wide Web in which the semantics of information and services on the Web are being defined. This is enabling the Web to understand and satisfy the requests of people and machines to use the Web content. Knowledge is an essential part of most Semantic Web applications and ontology, which is a formal explicit description of concepts or classes in a domain of discourse, is the most important part of the knowledge. Therefore, rule acquisition is the process of gaining knowledge or rules from similar websites. The rule acquisition is also an important issue, and the Web that implies inferential rules be a major source of rule acquisition. 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. Rule acquisition is as essential as ontology acquisition; still however rule acquisition is at a standstill a bottleneck in the operation of rule-based systems. This is time consuming and difficult, because it wants knowledge proficient as well as domain specialists, and in attendance are statement problems among them. Nevertheless, little bits rules have previously been implied in Web pages, and it is potential to acquire them from Web pages in the same manner as ontology learning.

The ontology can diminish the quantity of information and decrease the effort of utilizing the information in rule acquisition, since it is simplified and purposely reorganized for rule acquisition. In addition, the ontology can be accrued and reclaimed throughout replicated rule attainment. The ontology demonstration is simply unstated and maintainable; combining these attributes with the achievement domain-specialists have had in building libraries of ontologies across different regulations and the reality of a lot of tools which use ontology as the base demonstration.
Even though several of the prior representations are proficient of assuring the subsequent requirement of accomplishing difficulty solving during the use of procedural programming we coveted something that reminds you of the method specialists solve problems lacking commanding an additional algorithmic complexity. Professionals are inclined to solve problems through exploring their knowledge base for an appropriate solution. The ease of rule-based demonstration has completed it the majority normally used form for knowledge base systems. The other benefit this structure of representation has is its modularity as rules can be included or eliminated independently of other rules.

Ontologies capture the structure of the domain, i.e. conceptualization. This includes the model of the domain with possible restrictions. The conceptualization describes knowledge about the domain, not about the particular state of affairs in the domain. In other words, the conceptualization is not changing, or is changing very rarely. Ontology is then specification of this conceptualization - the conceptualization is specified by using particular modeling language and particular terms. Formal specification is required in order to be able to process ontologies and operate on ontologies automatically. Ontology describes a domain, while a knowledge base (based on ontology) describes particular state of affairs. Each knowledge based system or agent has its own knowledge base, and only what can be expressed using ontology can be stored and used in the knowledge base. However, sometimes rules have already been implied in Web pages, and it is possible to acquire them from Web pages in the same manner as ontology learning. That is, most of the rule components already exist in Web pages. It means that we can acquire rules more easily by using an automatic rule acquisition method rather than the old method with domain experts and knowledge experts. However, there are some problems with extracting rules from text for example, item is a variable and book is a value in. There are numerous possible combinations of making rules. Our idea for solving these problems is using rules of similar sites in limited situations under a couple of assumptions.

The main objective of these researches is to propose a rule acquisition procedure that automates repeated rule acquisition from similar sites by using the rule ontology RuleToOnto. This paper proposed a screening method and two main steps of rule acquisition, which consists of rule component identification and rule composition with the identified rule components.

Also we have proposed an optimization algorithm called memetic algorithm which serves as an effective intensification mechanism that is very useful when using sophisticated representation schemes and time-consuming fitness evaluation functions. These algorithms also incorporate a population, which gives them an effective explorative ability to sample huge search spaces. Another important aspect that has been investigated when designing memetic algorithms for scheduling and timetabling problems is how to establish the right balance between the work performed by the genetic search and the work performed by the local search. The main contribution of the work is first we are collecting the general rules from the various websites and given as input to the RuleToOnto algorithm. Using the RuleToOnto the Rule drafts will be created based on the identification of the variable and values and then compose rules criteria, and also the “if then else” condition will be checked which will be a main part. Then the BFS (Best First Search) algorithm is implemented to generate the best matched rules as forming the rule draft’s variable instances as a tree structure. If a variable instance is closer to the already chosen instances of a rule than the other instances of the same variable, we assign the instance to the rule. This geographic assumption plays a very important role in our approach. From these outcomes the best solution will be collected by using the optimization algorithm called Memetic algorithm. The overall research formed as follows: In section 2 the related work of the knowledge acquisition methods discussed. In section 3 the main part of RuleToOnto and the best first search is explained. In section 4 the memetic approach is discussed briefly. At last the experimental results are explained with that the conclusion and future work in section 5.
That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing. The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The first steps in weaving the Semantic Web into the structure of the existing Web are already under way [2]. The resulting infrastructure will spur the development of automated Web services such as highly functional agents. The challenge of the Semantic Web, therefore, is to provide a language that expresses both data and rules for reasoning about the data and that allow rules from any existing knowledge-representation system to be exported onto the Web. Learning by examples is a very different concept from rule acquisition from texts, which imply IF-THEN rules. Therefore, it is impossible to apply those methods in this paper problem, because their target is structured data while there target is unstructured text. Compared to rich studies of ontology learning, rule acquisition from the Web is not popular. Moreover, acquired rules are limited to a certain purpose and type [3], [12], and are not general-purpose inference rules. Most significantly, studies about automatic rule acquisition from text are quite rare while there are some studies that discover rules from existing data.

Even though these can be separated by the Related Works section into ontology learning and rule acquisition, the extraction of rules is one of the research areas in ontology learning, because the inference rules could be an outcome of ontology learning. The term “inference rule” means the relationship between two phrases in entailment rule approaches. Moreover, the rules are generated with statistical methods by calculating frequencies and probabilities while the rules are directly generated from the Web in this approach.

The eXtensible Rule Markup Language (XRML) approach is a framework for extracting rules from texts and tables of Web pages [4]. The core of the XRML framework is rule identification, in which a knowledge engineer identifies various rule components such as variables and values from the Web pages with a rule editor [4]. The effectiveness of the rule acquisition procedure of the XRML approach depends on the rule identification step, which also depends on the large amount of manual work done by the knowledge engineer.

Semantic Web Rule Language (SWRL) was designed to be the rule language of the Semantic Web. SWRL is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language the Unary/Binary Data log sublanguages of the Rule Markup Language. SWRL allows users to write Hornlike rules expressed in terms of OWL concepts to reason about OWL individuals. The rules can be used to infer new knowledge from the existing OWL knowledge bases [5].

The algorithm builds the taxonomy with linguistic analysis and identifies relevant candidates of classes and instances based on statistical analysis [6]. The Ontologies are composed from automatically obtained taxonomies. Some approaches used somewhat different learning methods for identifying instances and relations. For example, WEBfiKB [7] used Bayesian and First Order Logic learning methods, and Sanchez and Moreno [8] suggested a knowledge acquisition technique that built ontologies with a multiagent system. TextOntoEx[9] defined and used semantic patterns to identify not only simple taxonomic relations but also non taxonomic conceptual relations (e.g. causes, caused by, treat, contain, etc.). The approach using the Multiple Classification Ripple-down Rules (MCRDR) methodology [10], in ontology, learning is somewhat similar to this approach in its framework. They use a graph search algorithm instead of MCRDR to extract inference rules. In addition, they accumulate the rule ontology by repeating rule acquisition across different sites. The existing Rule acquisition is still a bottleneck in the deployment of rule-based systems. The existing system is using knowledge experts as well as domain experts. Consume more time also difficult to extract the rules from the web pages. It is also need knowledge experts and domain experts and the contact problems between knowledge experts and domain experts.

III. Extract Rule From Websites Using RULETOONTO

All para First, select the particular domain and select the Input Rule webpage from the particular domain. Then acquire the rule from input webpage. From the webpage, the rule can automatically extract by using screening method and display the webpage rule. Get the source from the rule webpage and identify variables and values.
A. Ontology Construction

After acquired the rules from webpage, then this article can construct the ontology based on identified variables and values. In addition, get another similar rule webpage related to input webpage by using this ontology information, there can be identify rule component in new input site.

B. Rule Component Identification

This article expanded RuleToOnto by adding synonyms of each term using WordNet. In the comparison between the terms of RuleToOnto and the terms of the Web page, if there is used semantic matching instead of simple string comparison. In order to find the semantic similarity between two terms used WordNet. Identified components are denoted in the format of variable instances with variable abbreviation and number. If there are, rules acquired from Amazon. Com (in short Amazon), it can make an ontology which shows the variables and values used in the rules. By using the information, there can be identify rule components in a new site such as Barnes&Noble.com. Semantic Similarity = \( \frac{1}{\text{path_length}} \)

This measure is calculated only when one term is a hyponym of the other term, and the path length is the path length between the two terms in the hyponym hierarchy. There are decided that two terms are semantically related when the measure is larger than 0.25.

C. Preparation & Ordering

The main objective of rule composition is to combine identified variable instances into rules. There are several possible variable instances for one variable on a Web page. The first step of rule composition is the preparation step, where to find appropriate rules from RuleToOnto. This is done by comparing the identified variable instances with the variables of the rules in RuleToOnto. The first job of preparation is extracting rule candidates from RuleToOnto. Every variable of each rule candidate should be matched to the variable instances of \( \text{VI} \). Number of all instances of each variable count is calculated. The objective of rule ordering is to decrease the complexity of making rules with identified variable instances. Therefore, calculate the number of possible combinations of assigning variables for each rule. The distance among a set of instances in a text can be calculated with a variance of instance positions. A low value of variance means that the instances are gathered around one place in the text. That is, the variable instance with the lower variance is more suitable for a rule instance.

This paper evaluate the distance between a variable instance and the already assigned instances to with the function which is very similar to the variance and by assigning the variable instance to rule candidate. \( \text{Dist}(\text{VI}_i, \text{RI}) = \frac{1}{n} \left( \text{Pos} (\text{VI}_i) - \mu^2 \right) + \frac{1}{n} \left( \text{Pos} (\text{VI}_1) - \mu^2 \right) + \cdots \left( \text{Pos} (\text{VI}_n) - \mu^2 \right) \)

Where the position of instance is text and \( \mu \) is Average.

The Best-First Search (BFS) is used in the evaluation function. The Initialization step includes choosing candidate rules, rule ordering, and variable ordering. When the algorithm succeeds in assigning variable instances to every variable of TotalOrder (RC), the loop ends and prints the path for the currentVI. It is a list of recommended rule instances.

D. Rule Refinement

Once the rules are determined, the next step is to complete the rules by assigning variable and value pairs to IF or THEN. The identified rule instances can be converted to the variable-value pairs by matching variables and values with identified values and the ontology. Assigning the pairs to IF or THEN is very simple. If the variable belongs to an IF part in the rule instance of RuleToOnto, this can be assign the pair to the IF part of the rule. Otherwise, if it belongs to a THEN part, this paper assigns it to the THEN part. The rules automatically generated are not complete in most cases, so they need to refine them. The knowledge engineer checks the rules and modifies connectives and values. [20]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.

IF days_of_shipment <= 40
AND return_status = "original condition"
AND item = "book"
THEN
Refund = "full"

IV. MEMETIC ALGORITHM FOR OPTIMAL RULE ACQUISITION

It is generally believed that memetic algorithms are successful because they combine the explorative search ability of recombinative evolutionary algorithms and the exploitive search ability of local search methods. An analogy is that the evolutionary part of a memetic algorithm attempts to simulate the genetic evolution of individuals through generations, while the local search part attempts to simulate the individual learning within a lifetime.
The majority of memetic algorithms proposed in the literature are a result of incorporating some form of local search to a genetic algorithm. This is illustrated in Fig. 1. This local search can be for example, constructive heuristics, repair methods, specialized self-improvement operators, etc. The local search phase can be applied before, after or in between the genetic operations. However, as discussed in [13], the interaction between the memes and the genes can be even more sophisticated than that and most implementations of memetic algorithms fail to reflect the complex interactions of the memetic paradigm. Krasnogor [13] argues that in a truly memetic system:

1. Memes also evolve representing the way in which “individuals learn, adopt or imitate certain memes or modify other memes” and,
2. The distribution of memes changes dynamically within the population representing the effects of “teaching, preaching, etc.” inside the population of people.

There are units a substantial range of applications of memetic algorithms to programming issues reportable within the literature including: machine programming, academic timetabling, personnel programming, maintenance programming among several others.

As mentioned above, the addition of local search helpers into genetic algorithms is the most common approach reported in the literature. The variety of helpers that have been proposed range from the use of tailored chromosome representations and operators and simple repairing methods based on constraint-based reasoning to very sophisticated combinations of algorithm components in which a memetic algorithm is embedded into a genetic algorithm. Depending on the success of the operator, they calculate an average growth value which is used to dynamically adjust the probability of each mutation operator. More specifically:

1. When a mutation operator M is applied, a solution M(x) is generated from a solution x.
2. The progress of the mutation operator M when applied to solution x is 1 if x is dominated by M(x), 0 if x dominates M(x) and 0.5 otherwise. A solution x dominates a solution y if x is as good as y in all objectives and better in at least one of them [14].
3. The average Progress(M(i)) of each mutation operator M is calculated by summing all the progresses of M and dividing it by the number of solutions to which M was applied.
4. Then, the probability of each mutation operator is adjusted using (1) where η is the number of mutation operators and δ indicates the minimal ratio value permitted for each operator. That is, δ is a parameter that permits to keep each operator even if the progress of the operator is too poor. \( P_M(i) = \frac{\text{Progress}(M(i))}{\sum_{j=1}^{\eta \text{Progress}(M(j))}} \times (1 - \eta \times \delta + \delta) \)

As discussed here, although some research has been carried out on how to adapt the application of different genetic and local search operators and heuristics (memes) throughout the search, in most cases the memes have been designed before the search and remain unchanged during the process. The notion of evolution of memes instead of evolution of genes in the context of timetabling was first suggested by Ross et al. who said: “we suggest that a GA might be better employed in searching for a good algorithm rather than searching for a specific solution to a specific problem” [15]. As also argued by Krasnogor, the evolution of memes is an aspect that deserves more attention in order to design more advanced and improved memetic systems [16].

---

**Fig. 1. Memetic Algorithm**
V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Execution Time/Sec

This graph contains execution time of proposed and existing system. The execution time of existing time is very high compared with the proposed system. Using ontology in this approach automate the rule acquisition procedure. The initial point of this approach there using a screening method it will be helpful for automatically acquiring rule from a site, and acquired similar rules from other similar sites of same domain automatically. Rule ontology, can be use in this paper, it includes the information about the rules including terms, rule component type and rule structures. This shown in Fig 2.

B. Accuracy

The above graph shows that the accuracy of proposed system is high. In proposed system, this paper automatically extract rules from websites then the accuracy is also very important. RuleToOnto represent the IF and THEN part of each rule by connecting rule with variables with the IF and THEN relation. If there are some information about the variables and values, and the connection between the variables and values the instance variable must have at least one value instance and rule have at least one variable for each IF and THEN properties. This shown in Fig 3.
VI. Conclusion

Ontology is used to propose an automatic rule acquisition procedure, named RuleToOnto that includes information about the rule components and their structures. This paper started from the idea that it will be helpful to acquire rules from a site if we have similar rules acquired from other similar sites of the same domain. RuleToOnto includes information about the rule components and their structures. RuleToOnto is a generalized, condensed, and specifically rearranged version of the existing rules. The rule acquisition procedure consists of the rule component identification step and the rule composition step. Designing a memetic algorithm is frequently associated with the incorporation of knowledge from the problem domain in the form of helpers to evolutionary algorithms. We should be careful because almost every new piece of specific knowledge that is added to a memetic algorithm can potentially produce improved results. Then, we can many times keep designing a ‘new’ or an ‘improved’ version of the memetic algorithm, i.e. an incremental design of algorithms. We should focused on the main ideas and strategies without getting lost in the details of the different implementations.

REFERENCES


