Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Clustering rule bases using ontology-based similarity measures

Saeed Hassanpour, Martin J. O'Connor, Amar K. Das [*]

Stanford Center for Biomedical Informatics Research, MSOB X215, 251 Campus Drive, Stanford, CA, 94305, USA

ARTICLE INFO

Article history:
Received 30 June 2011
Received in revised form
31 March 2013
Accepted 13 March 2014
Available online xxxx

Keywords:
Rule management
Rule categorization
Semantic similarity
SWRL
OWL

ABSTRACT

Rules are increasingly becoming an important form of knowledge representation on the Semantic Web. There are currently few methods that can ensure that the acquisition and management of rules can scale to the size of the Web. We previously developed methods to help manage large rule bases using syntactical analyses of rules. This approach did not incorporate semantics. As a result, rule categorization based on syntactic features may not be effective. In this paper, we present a novel approach for grouping rules based on whether the rule elements share relationships within a domain ontology. We have developed our method for rules specified in the Semantic Web Rule Language (SWRL), which is based on the Web Ontology Language (OWL) and shares its formal underpinnings. Our method uses vector space modeling of rule atoms and an ontology-based semantic similarity measure. We apply a clustering method to detect rule relatedness, and we use a statistical model selection method to find the optimal number of clusters within a rule base. Using three different SWRL rule bases, we evaluated the results of our semantic clustering method against those of our syntactic approach. We have found that our new approach creates clusters that better match the rule bases' logical structures. Semantic clustering of rule bases may help users to more rapidly comprehend, acquire, and manage the growing numbers of rules on the Semantic Web.

1. Introduction

Rules have been a central form of knowledge representation since the earliest development of intelligent systems [1–5]. As it has been repeatedly shown in the Semantic Web Stack, rules are an essential layer of knowledge on the Semantic Web framework. In the last few years, rules are increasingly being used to represent and apply knowledge in a range of Semantic Web applications [6–10]. However, in contrast to other knowledge representation formats on the Semantic Web such as ontologies and taxonomies, rules do not have a hierarchical structure to facilitate their presentation and comprehension. Although some rule representation and interchange formats such as RuleML [11] and RIF [12] allow users to annotate rules through adding metadata, the flat structure of rules negatively effects the applications of rules on the Semantic Web and the use of rules to their full potential in this framework.

In many past and current rule-based systems, domain experts manually organize rules through labeling and assigning to different groups based on their semantics. For example, the Cyc project has an enormous number of rules, often referred to as *a sea of assertions*, in its comprehensive knowledge base of common sense knowledge for general-purpose reasoning [1]. To organize such a knowledge base, domain experts divide these rules into microtheories, which are collections of concepts in a particular realm of knowledge. These microtheories are used for domain separation, conflict resolution, and hypothetical reasoning. Although this manual categorization of rules is semantically precise, it is a laborious task. In general, we would like automated methods to manage rule bases when the number of rules becomes large and relationships among rules are too complicated for developers and domain experts to comprehend.

A related field to our work in this paper is association rule clustering. Association rules are used in data mining to present frequent data patterns in databases [13]. Consider a database of tuples, where each tuple is a set of attribute values, which are called items. An association rule is defined on these tuples as $A \rightarrow B$, where A and B are disjoint subset of items. For example $(age = 30) \rightarrow (own\_car = yes)$ is an association rule with one attribute value on left and another one on the right hand side of the rule. The clustering of association rules is combining similar association rules to form more general rules. Therefore, the clustered rules can cover a range of values for attributes. For instance, the clustered rule $(30 \leq age < 32) \rightarrow (own\_car = yes)$ can be generated from combining two association rules $(age = 30) \rightarrow (own\_car = yes)$ and $(age = 31) \rightarrow (own\_car = yes)$.

[*] Corresponding author. Tel.: +1 650 736 1632; fax: +1 650 725 7944.
E-mail addresses: saeedhp@stanford.edu (S. Hassanpour), martin.oconnor@stanford.edu (M.J. O'Connor), amar.k.das@dartmouth.edu, amar.das@stanford.edu, saeedhp@gmail.com (A.K. Das).

http://dx.doi.org/10.1016/j.websem.2014.03.001
1570-8268/© 2014 Elsevier B.V. All rights reserved.

Lent et al. [13] introduced a method to preform association rule clustering over two attribute values in a database. In this method the values for two attributes in a database are presented as a two-dimensional grid, and a geometric-based greedy algorithm is used to find large homogeneous regions in the grid. These regions represent the approximate optimal clusters of association rules for those two attributes. Other techniques to cluster association rules include decision trees. The decision tree methods, such as CART [14] and C4.5 [15], provide greedy algorithms to find an approximate model for the data by finding homogeneous regions in a multi-dimensional space. These models also can be used to predict the value of a target variable based on several input variables.

A major difference between association rules and the rules in our work is the fact that association rules usually refer to a single tuple in a database; therefore, an association rule can be considered as a single atomic proposition in the logic programming setting. However, SWRL rules usually refer to multiple individuals in a single rule, and thus contain more than one atomic proposition. As the result the association rules clustering methods that handle only one logical proposition are not applicable to SWRL rule clustering.

Furthermore, association rules are defined on attributes from a database schema and their corresponding values. In contrast, SWRL rules can contain object property atoms that relate one individual to other individuals rather than to other attributes. Therefore, there is a distinction between SWRL rule clustering and association rule clustering methods that only deal with combining values for attributes.

In another related work, an object-oriented similarity measure is presented to find the best Semantic Web service match for queries [16]. In this work the Web service profiles and queries are presented as instances of an OWL class. This work focuses on the properties that are associated to these instances. In this method, for data value properties, the matching of arguments' values or types is considered as their similarity. The similarity of object properties is determined by the shortest path length of their arguments' classes in an object-oriented model, which is based on the OWL class hierarchy. The aggregation of the similarity between data value and object properties of two instances is computed as their overall similarity. In contrast, in our work we are interested in measuring the similarity of two rules rather than two class instances. Furthermore, the mentioned method is designed for the applications that two compared instances are from a same class in an ontology [16]. However, in our work rules typically consist of diverse sets of classes and properties. In addition, our rule base analysis does not execute the rules. Therefore, object property arguments are not bound to any class instances. In this case, the available information about the object properties' arguments is limited to their domain and range. These domains and ranges, according to the rule bases in our evaluation, are usually too general to extract a similarity between properties.

Rule clustering has also been used in rule systems to provide the parallel execution of rules and expedite their running-time. For example CREL rule system performs a compile-time rule dependency analysis to partition the rules into independent clusters [17]. The rules in different clusters can be executed in parallel during the rule base execution. In another work in this domain, rule base parallelization is facilitated by introducing a new rule language, PARULEL [18]. Using this language the programmers are able to provide instructions for rules' parallel execution through meta-rules. In these methods the focus of rule clustering is providing run-time parallelization, and is different from our work's purpose, which is facilitating understanding the rules and their semantics in rule bases.

Another rule clustering method is motivated by optimizing the execution of Event–Condition–Action (ECA) rules in object-oriented active knowledge base systems [19,20]. ECA rules are composed of three sets of events, conditions, and actions. If the condition part of all the events that are specified in the event parts of the rules hold, the action parts will be executed. In practice, the presence of multiple ECA rules with the same action part causes multiple inefficiencies, such as difficulty in rule maintenance and redundancy in condition checking and action execution of the rules. Also this many-to-one relationship between events–conditions and actions leads to a problem with the execution synchronization of multiple rules due to the immediate events–conditions coupling in active knowledge base systems. This problem is known as the net effect [20]. As a solution to these problems, ECA rules with similar action parts are translated to a single rule with more complex events. These new rules can be considered as the clusters of pre-existing rules. In this method rules are grouped in different clusters, where each cluster performs the same task in an object-oriented active knowledge base system. In contrast, our rule clustering method in this paper is based on rules' semantics rather than optimizing their run-time performance.

Rule management methods are widely adopted in the business domain to assist users in the rule management challenge [21]. These tools typically provide high-level rule management interfaces and editors, and support for multiple data models for rules, user-friendly rule presentation, acquisition, functionality, and rule testing and refinement methods. Other features include rule versioning, access control, justification, rule argumentation, and real-time debugging [21–23]. Although these management techniques can be useful, they are typically restricted to their particular business domain.

In our experiences in developing general methods for rule management for SWRL rule bases, we have found that when the number of rules exceeds a few dozen it becomes extremely hard for domain experts to comprehend the content of the rule base and the relationships among the rules [24–26]. In a recent work, we leveraged the syntactic structure of SWRL rules to categorize, visualize, and paraphrase them [25]. The analysis was used to generate an abstract presentation for each rule, called a rule signature, which we used to categorize rules. Although this method was successful in revealing the structural patterns among the rules, it could group rules that do not have similar semantics together. In practice, categorization results were often coarse. This approach is also relatively brittle: a syntactic change in a rule that had no change in its logical assertion could cause a rule to be switched to another grouping.

In this paper, we present a rule categorization method for ontology-based rule languages, such as SWRL and Cyc. Our approach uses ontology-based relationships among rule atoms to automatically partition rule bases into semantically similar clusters. In this work, we use multiple measures of semantic similarity and investigate the impact of the semantic similarity measures on the rule categorization results. In our approach, we focus on SWRL [27], which is the primary language for encoding rules in OWL [28]. SWRL rules can be considered as formal logical statements about entities in an OWL ontology. Because all referred entities in SWRL rules, are also presented in the associated OWL ontology, automated techniques can use ontology-based relationships in rule clustering. In evaluating the accuracy of our rule clustering method, we compared our method against the syntactic approach on three biomedical ontologies that contain SWRL rule bases.

2. Background

Ontology-based rule languages provide an opportunity to perform a semantic analysis of rules based on relationships encoded in the ontologies. Direct references between rule atoms and their relationships in ontology hierarchies improve the understanding of the relationships among rules. For example, a rule about hypertension can also be inferred to relate to rules about blood pressure.