Rule identification using ontology while acquiring rules from Web pages

Sangun Park\textsuperscript{a,}\textasteriskcentered, Jae Kyu Lee\textsuperscript{b,c}

\textsuperscript{a}Division of Business Administration, Kyonggi University, San 94-6 Yiui-Dong, Paldal-Gu, Suwon, Kyonggi 442-760, South Korea
\textsuperscript{b}Graduate School of Management, Korea Advanced Institute of Science and Technology, 207-43 Cheongryangri-dong, Dongdaemun-gu, Seoul 130-012, Korea
\textsuperscript{c}School of Information Systems, Singapore Management University, 80 Stamford Rd, Singapore 178902, Singapore

Abstract

As research on the Semantic Web actively progresses, a more intelligent Web environment is expected in various domains including rule-based systems and intelligent agents. However, rule acquisition is still a bottleneck in the utilization of rule-based systems. To extract rules from Web pages, the framework of eXtensible Rule Markup Language (XRML) has been developed. XRML allows the identification of rules from Web pages and generates rules automatically. However, the knowledge engineer’s burden is still high because rule identification requires considerable manual work. In order to reduce the knowledge engineer’s burden, we proposed an ontology-based methodology of enhanced rule identification. First, we have designed an ontology OntoRule for automated rule identification. Also, we proposed a procedure of rule identification using OntoRule. Lastly, we showed the performance of our approach with an experiment.

The rule acquisition framework based on eXtensible Rule Markup Language (XRML) (Lee and Sohn, 2003) has been developed in order to acquire structured inferential rules from Web pages. The main objective of this paper is to propose an enhanced method of rule acquisition based on the XRML framework by adopting an ontology. The acquired rules should be utilized in the reasoning of rule-based systems.

1.1. Rule acquisition with XRML

The XRML approach is a framework for extracting rules from texts and tables (Kang and Lee, 2005). XRML consists of three components as shown in Fig. 1: Rule Identification Markup Language (RIML), Rule Structure Markup Language (RSML) and Rule Triggering Markup Language (RTML) (Lee and Sohn, 2003). RIML identifies the rules implicitly expressed in Web pages, RSML represents the formal rule structure which corresponds to the rule syntax in commercial rule-based systems; and RTML defines the conditions that trigger the inference of certain rules (Lee and Sohn, 2003).
The rule acquisition and utilization procedure of the XRML approach is illustrated in Fig. 1 with the step numbers in parentheses (Kang and Lee, 2005). Steps 1–4 are rule acquisition steps. In step 1, the knowledge engineer selects the relevant Web pages that contain target rules. In step 2, s/he identifies various rule components from the Web pages with a rule editor (named XRML Editor), and the identified result is saved in RIML files. Rule components are variables, values, IF/THEN parts, and connectives that constitute target rules. Step 3 is an automatic generation of an RSML draft by the XRML Editor. The generated draft rules may be incomplete. Therefore, the knowledge engineer should refine draft rules to build a complete rule set in step 4. Step 5 is the utilization of acquired rules for human and software agents. If any changes happen on the Web pages or rule bases, we can detect their counterparts by the link between RIML and RSML to apply the changes.

Fig. 1. Rule acquisition process by the XRML approach (Kang and Lee, 2005).

1.2. Research objectives

If a knowledge engineer repeats rule acquisition over similar sites of the same domain, s/he may be able to identify required rule components by using experience of the previous sites. We expect that a well-designed ontology can substitute this experience. The ontology includes information on rule components and structures acquired from another site in the same domain. By using the ontology, the XRML approach can be extended to automatically identify rule components such as variables, values, and rules from Web pages.

Therefore, the first objective of our research is to show the role of the ontology in rule acquisition and address various issues on using the ontology (Section 3). The second is to design an ontology for rule acquisition (Section 4). Once the ontology is designed, the next objective is to propose an enhanced rule identification procedure using the ontology in the XRML approach (Section 5).

In the case study of our approach, we adopt a shipping rates and returns policies domain of real world online bookstores (Amazon.com, Barnes&Noble.com, and Powells.com) to show various examples of the ontology and rule acquisition. By acquiring rules on shipping rates and returns policies from bookstores, it is possible to build a comparison shopping system which compares not only book price but also shipping cost and returns policies based on the rule-based system. We will show the performance of our approach through an experiment with the rules in Section 6.

2. Review of ontology and knowledge acquisition

In this section, we review the literature on ontology, rule markup language, and knowledge extraction based on ontology.
2.1. Ontology, rule markup language, and knowledge acquisition

Ontology is commonly explained as a specification of domain knowledge conceptualization (van Heijst et al., 1997). Therefore, an ontology provides basic knowledge which allows applications to understand the various data of the Web (Chan and Lam, 2005). Most research represents an ontology with standard ontology languages such as RDF(S) (Manola and Miller, 2004; Brickley and Guha, 2004), DAML+OIL (Connolly et al., 2001; Horrocks, 2002), and OWL (Smith et al., 2004). Also, there are various tools to support building, editing, browsing, and searching an ontology (Dzbor et al., 2003; Brockmans et al., 2004; Knublauch et al., 2004; Liebig and Noppen, 2004). Since the Web has rich sources for ontology, there have been increasing efforts related with the Semantic Web to automatically extract an ontology from the Web (Maedche and Staab, 2000; Hemnani and Bressan, 2002; Lebbink et al., 2002). Many automatic or semi-automatic ontology extraction systems rely on pre-defined templates and pattern-based extraction rules or machine learning techniques to identify entities in text documents (Babowal and Joerg, 1999; Craven et al., 2000; Vargas-Vera et al., 2001; Ruiz-Sánchez et al., 2003).

Recently, ontology-based reasoning (Donini et al., 1996; Crow and Shadbolt, 2001; Horrocks, 2002; Kopena and Regli, 2003) based on description logic (Donini et al., 1996; Baader and Nutt, 2003) has become a popular issue related to the Semantic Web. However, ontology-based reasoning has limitations compared with rule-based reasoning, and ontology is not sufficient to represent inferential knowledge (Harmelen and van Fensel, 1999). The Rule Markup Initiative (RuleML, 2003) emerged to handle rules beyond ontology. The goal of the Rule Markup Initiative is to define standard syntax for the exchange of compatible inferential rules in XML. As the result of their efforts, the proposal for the rule representation standard based on ontology is progressing as SWRL (Horrocks et al., 2004). SWRL represents Horn-like rules by extending the axiom of OWL to integrate OWL with RuleML.

However, most research of the RuleML community is still focusing on rule representation rather than rule acquisition from the Web. In order to utilize vast resources of knowledge in the Web, research on rule acquisition from the Web is essential. In this regard, our approach focuses on rule acquisition from Web pages.

2.2. Knowledge extraction based on ontology

In addition to the research on building an ontology from the Web, research using ontology in knowledge acquisition from the Web is also increasing. One approach of this research is annotating documents by using ontology (Mukherjee et al., 2003; Kiryakov et al., 2004; Popov et al., 2004). The goal of this approach is to make machines understand the theme of documents so that the software classifies, searches for, and retrieves required documents. For example, Kiryakov et al. (2004) and Popov et al. (2004) used an ontology to organize and search for documents under various conditions and interests. Mukherjee et al. (2003) also used an ontology to automatically transform template-based content-rich HTML documents into their corresponding Semantic partition trees by using ontology. However, this kind of approach is focused on the classification of information using an ontology rather than the extraction of general knowledge from documents.

On the other hand, ontology can be used to extract general knowledge from Web pages. In this kind of approach, an ontology is used as a predefined template on knowledge extraction (Vargas-Vera et al., 2001; Alani et al., 2003). For example, a user may annotate knowledge on the Web page using the template of an ontology through an annotation tool (Vargas-Vera et al., 2001), or a knowledge extraction program may automatically retrieve required knowledge to fill the ontology template (Alani et al., 2003). In the process of automatic extraction of knowledge, WordNet (Miller, 1995), which is a general-purpose lexical database, and GATE (Bontcheva et al., 2004) are used to parse Web pages.

In other research, an ontology is utilized for the automatic acquisition of domain models for constraint-based tutors in intelligent tutoring systems (ITS) (Suraweera et al., 2004). In this approach, an author should build a domain ontology before acquiring domain models. The ontology not only automates acquisition of domain models, but also enhances the author’s understanding of the domain, even if the procedure is not efficient because the ontology is not reused or shared.

In our approach, we try to extract inference rules rather than factual and terminological knowledge from Web pages. Furthermore, we design a new ontology and use the ontology in rule acquisition. The ontology in our approach plays a similar role to ITS (Suraweera et al., 2004) in the sense that it automates some procedure and helps the knowledge engineer to understand the domain. Furthermore, in our approach, the ontology can be accumulated, shared, and reused to reduce the burden of ontology building in each rule acquisition.

2.3. Mining association rules from text

Association rules, which show correlations between keywords in the texts, are easy to understand and to interpret for an analyst (Feldman and Dagan, 1995). The difference of applying association rules in a text framework from applying them in databases is the special characteristics of text as unstructured data (Delgado et al., 2002). The basic idea of the approaches is that humans tend to use language to generate texts from predefined structures and not from separate single words. Since co-occurrences of words are reflected in the sentences of the language, they can be observed and extracted (Perrin and Petry, 2003). Perrin and Petry suggested algorithms to find collocational
expressions that represent effectively the significant content of the text. An automatically extracted collocational expression is formed with content words automatically recognized by the algorithm. The approach is suitable for knowledge discovery tasks. However, it may not be suitable for tasks requiring deeper understanding using the actual meaning of each phrasal component (Perrin and Petry, 2003). Rule acquisition from Web pages is an example of those tasks.

Approaches of mining association rules from text have much in common with our research in extracting rules from text. However, the purpose of rules in text mining is different from ours. The objective of text mining is to extract relevant facts from a text and assign each with a theme label, but our objective is to extract inferential rules for reasoning in rule bases systems. For example, we may extract association rules of collocational expressions related to shipping rates in our sample domain of shipping rates and returns policies. However, the association rules cannot be used in reasoning for the calculation of shipping rates because the information is relations between words rather than inferential rules. Nevertheless, the association rules are useful to find rule components such as variables and values from Web pages as the rule components are related to each other and coupled to the domain. Therefore, we expect that the integration of text mining and our research will be an interesting future research issue since building ontology by using text mining is independent from rule acquisition.

3. Overview of ontology-based rule identification

This section describes the role of ontology in rule identification and the basic idea of rule identification using an ontology based on XRML. We also describe various issues on using ontology in rule identification and assumptions of our approach.

3.1. Role of ontology in rule identification

As described in Section 1, the knowledge engineer identifies basic rule components from Web pages in the rule identification step. To achieve this, RIML version 1.0 (Lee and Sohn, 2003) defines the tags like `<RuleGroup>`, `<Rule>`, `<variable>`, and `<value>`. For instance, suppose we have picked the Web page of Barnes&Noble.com, as shown in Fig. 2, which explains the returns policies of books. If the knowledge engineer recognizes that `books` and `CDs` in Fig. 2 are values, s/he can represent the rule components by using tags such as `<value>books</value>` and `<value>CDs</value>`, respectively. In RIML version 2.0 (Kang and Lee, 2005), the tags like `<RuleTable>`, `<IF>`, `<THEN>`, `<AND>`, `<OR>`, `<NOT>`, and operators were added to increase the degree of rule components representation.

In the experiment that acquires delivery rules from Amazon.com, Barnes&Noble.com, and Powells.com, 97.7% of total rules and 88.5% of total rule components were found in the identification step (Kang and Lee, 2005). It proves that the rule identification step is very important and decisive among all rule acquisition steps, because the more components we identify, the more rule drafts the XRML editor automatically generates. However, rule identification still highly depends on the knowledge engineer’s manual work.

Manual work consists of finding variables and values which constitute inference rules from the Web page, associating them to make pairs, and assigning the pairs to rules. Knowledge and experience about domain and rules are needed for the knowledge engineer to accomplish those manual tasks. A well-defined ontology can substitute for the knowledge and experience, and can reduce the knowledge engineer’s manual work. The rule identification procedure can be automated by using the ontology.

3.2. Using ontology in rule identification

How can we use an ontology in rule acquisition? Let us assume that we have an ontology that was generated from Amazon.com as shown in Fig. 3 and we use it in rule acquisition from a Web page of Barnes&Noble.com as shown in Fig. 2. From the ontology as shown in Fig. 3, we can recognize that `refund` and `Return within 30 days` in Fig. 2 are variables and `books`, `CDs`, and `VHS tapes` are...
values. We can identify those words for variables and values from the ontology.

The ontology can provide more information for rule acquisition. For example, it can help to identify omitted components from the Web pages. We can perceive that *item* is omitted in Fig. 2 because *books*, *CDs*, and *VHS tapes* are values of *item* in Fig. 3. The ontology can also help to identify *IF* or *THEN* parts of rules. The detailed procedure of rule identification is described in Section 5.

Where can we obtain the ontology? The basic idea of obtaining an ontology is reusing rules that were acquired in the previous Web site. Therefore, we need a seed rule base in our approach. The various scenarios of using an ontology will be discussed in Section 3.3.2. The ontology in Fig. 3 can be generated from rules of Amazon.com. It is straightforward to extract the ontology from the following rule:

```
IF "Return within 30 days" = No
   OR (Item = BOOK AND "Returned status" = "Obvious signs of use")
   OR ((Item = CD OR Item = DVD OR Item = "VHS Tape")
       AND "Returned status" = Opened)
   OR "Returned status" != "Original condition"
THEN Refund = Partial
```

### Fig. 3. Example of ontology acquired from Amazon.com.

### 3.3. Issues on using ontology in rule identification

We explained the basic idea of using ontology for rule identification in the previous section with the example of a specific ontology. In this section, we discuss several issues in using ontology in rule identification. By considering the issues, we can build basic frames of ontology design and the automatic rule identification procedure.

#### 3.3.1. Specific ontology for rule acquisition vs. external ontology

The first issue in using ontology in rule identification is the ontology type in our approach. First, we can use a general-purpose lexical database such as WordNet (Miller, 1995) for an external ontology as other knowledge extraction approaches do. There are benefits of using an external ontology such as finding words for variables and values related to the domain and finding synonyms. However, this kind of help is merely primitive. It is hard to decide whether the words are variables or values, and associate variables with appropriate values. The goal of our research is to provide not only vocabulary and the relationships between the words but also information on the rule components, the role of each component in rules, and the structure of rules. Therefore, we will define a new ontology that can be generated from acquired rules. Let us call this ontology *OntoRule*.

#### 3.3.2. The ontology learning effect on the repeated rule acquisition

An ontology may be refined with the repeated rule acquisition from various sites of the same domain. There are various scenarios involving the use of ontology in the rule identification procedure. The simplest scenario is to acquire an ontology from a Web site and use the ontology in the same site. The ontology is not reused in this scenario. The second scenario is to acquire an ontology from a site and use the ontology in other sites of the same domain. However, the ontology may not be accurate and effective in other sites because the vocabularies and structures of rules are different from other sites. However, the ontology may not be accurate and effective in other sites because the vocabularies and structures of rules are different from other sites.

The third scenario is to modify the ontology when we apply it in other sites. We can accumulate vocabularies, synonyms, and structures of rules from multiple sites. As the result of accumulation, we expect the accuracy and effectiveness of the ontology to increase.

#### 3.3.3. A top–down approach vs. bottom–up approach in rule identification

A top–down approach in rule identification starts from identifying larger components such as *RuleGroup* or *Rule* and proceeds to smaller components such as *IF*, *THEN*,...
variable, and value. When the knowledge engineer identifies rule components without an ontology, the top–down approach may be useful. However, it is not appropriate for automatic recommendation of rule components using an ontology because the information is not sufficient for identifying rules in the initial step of rule acquisition.

A bottom–up approach starts from variable and value identification, and proceeds to Rule. Recommending variables and values from the ontology is more desirable in the initial step. For example, books, CDs, and VHS tapes in Fig. 2 are easily recommended as values from the ontology. As more variables and values are identified, OntoRule can accumulate enough information to identify IF, THEN, and Rule. Therefore, we adopted the bottom–up approach in the enhanced XRML editor (named OntoXRML).

3.3.4. Management of the ontology range in the rule identification procedure

Another issue of rule identification using ontology involves performance. We can estimate the performance of the proposed rule identification method using missed recommendations and wrong recommendations. If the ontology is sufficient in its vocabulary, missed recommendations will decrease, although wrong recommendations may increase because the large ontology may recommend more inappropriate variables and values. Consequently, there is a trade-off between missed recommendations and wrong recommendations. To control this problem, we control the range of the ontology used in rule identification. The range means a set of variables, values, and rules including IF and THEN parts, which are related with a topic. An ontology with a broader range means a bigger set of more variables, values, and rules.

The knowledge engineer can narrow down the ontology range in the order of domain, application, and rule group. For example, an ontology of the domain level range includes several ontologies of the application level range. In Section 4, we designed an ontology which includes such multi-level range.

3.3.5. Detection order of variable and value

Where should we start to detect rule components from a Web page, variable or value? First, a set of values are linked to a variable in the ontology. So if we detect a variable first, we can reduce the search range of value candidates to the linked values of the detected variable. Second, the number of variables is much smaller than the number of values in the ontology. Therefore, starting from a variable is better to decrease complexity. Third, if we know the type of variable, it is helpful to find values linked to the variable. For example, with a variable Item, which is Character string type and has values like Book, CD, and DVD in the ontology, we can search for only those values in Web pages. Besides the variable type, the ontology can include other tips which help automatic rule identification.

3.3.6. Detection of omitted variables

Identifying omitted variables is one of the most difficult things in rule identification. In Fig. 2, it is hard to identify that Returned Status, which is omitted in the Web page, is a variable for original condition. However, the ontology in Fig. 3 shows that original condition is the value of Returned Status. The identification of omitted terms significantly affects the degree of automation on RSML draft generation. In the experiment on the effect of identification (Kang and Lee, 2005), omitted terms occupied about 20% of the total terms.

3.4. Assumptions of our approach

3.4.1. Rule expressiveness

The acquired rules in our approach support backward chaining and forward chaining. Rules are represented by various combinations of conditions and conclusions for readability, and can be transformed into Horn clauses in the execution stage. In our approach, every atomic formula consists of a variable, a value, and an operator in RSML syntax (Kang and Lee, 2005). It has a tradeoff between expressiveness and ease-of-use. We have chosen ease-of-use and readability in the RSML format. Also, it reduces the complexity of detection of values, matching variables with values, and omitted variable detection in rule identification.

3.4.2. Target Web page

One of the assumptions of our research on a target Web page is that similar sites should exist for utilizing acquired ontology. Our approach works well in the domain where rules are repeatedly acquired from multiple sites which have similar rules. In our experiments, shipping rates and returns policies in bookstores are good examples. There are several domains which are desirable for our approach such as terms of agreement in various shopping malls, insurance rates and policies in insurance companies, and loan policies in banks. Another assumption is that the Web page should include practical and executable rules. In case of laws, which need interpretations by experts, it is hard to apply our approach directly.

3.4.3. Target application

The next assumption is that target application, which will use the acquired rules, requires rule-based reasoning. For example, an application for free shipping and returns policies needs reasoning rather than simple calculation. Our approach is desirable when Web pages are frequently updated. It can reduce the burden involved in system management.

3.4.4. Roles of the knowledge engineer

The knowledge engineer’s role in rule acquisition is still very important even though most parts of our approach are automatized. For example, selecting a proper range of Web pages for rule groups and rules depends on the knowledge engineer. Therefore, it is very important for the
knowledge engineer to understand the contents of a Web page.

4. Design of OntoRule for rule acquisition

This section describes the requirements of OntoRule and the details of OntoRule design. Ontology design includes representation of variables, values, rule structures, and attributes of variables and values for rule identification.

4.1. Requirements of OntoRule

We address requirements of ontology from the issues on using ontology in Section 3.3. Requirements are categorized into 5 types:

4.1.1. Information on the domain of rules for managing the ontology range

The knowledge engineer chooses a target Web page for rule acquisition at the beginning of the rule acquisition procedure. By maintaining information on domains and applications in OntoRule, the knowledge engineer can select a proper range of the ontology for rule acquisition.

4.1.2. Vocabulary and relationship information on variables and values of rules

To recommend appropriate words for variables and values from the Web page, OntoRule includes information on words that were frequently used as variables and values. Also, the relationships between variables and values are needed so that the XRML editor, OntoXRML can recommend proper association of values with variables.

4.1.3. Synonyms for domain specific terms

Another important function of an ontology is the representation of synonyms. WordNet (Miller, 1995) provides synonyms and coordinate terms for a given word, but there are domain specific synonyms in rule acquisition. We added these domain specific synonyms to OntoRule. In the experiment (Kang and Lee, 2005), 15% of total identified terms were synonyms.

4.1.4. Information on rule structure

To support identification of rule, IF, and THEN parts, the ontology includes information on rule structure which represents the relationship between terms and rules. Ontology for rule structure means generalized information on rules including IF and THEN parts. For example, several similar rules in a rule base can be generalized into one rule. Also, an IF part of a rule in the ontology does not include all components of the IF part of the rule in the rule base.

4.1.5. Additional information on variable and value

By adding variable type, various counts in rules, unit, and numeric range, OntoXRML can help to identify appropriate values for variables as described in Section 3.3.5.

4.2. Design of OntoRule

In this section, we discuss design of OntoRule based on the requirements described in Section 4.1. We use frame expression to represent our ontology design because frame expression is simple and easy to understand, and moreover the ontology with frame representation can be automatically and easily transformed to other ontology representation such as RDF and OWL.

Fig. 4 shows the entire ontology schema of OntoRule in frame representation. The requirements which we described in Section 4.1 are applied to the ontology schema. We define Domain, Application, and RuleGroup classes for ontology range management, and Rule class for rule structure. Also we define Variable, Value, and Synonym classes for the representation of variables and values. We will explain the details of the design in the following section with the ontology example from Amazon.com which is shown in Fig. 5.

4.2.1. Information on the domain and application for managing the ontology range

Managing the ontology range focuses on narrowing the range of the ontology which will be used in rule acquisition. We define Domain, Application, and RuleGroup classes to narrow down the range of the ontology in the order of Domain, Application, and RuleGroup as shown in Fig. 4. They are connected with the Applications and RuleGroups relations. The slot source is the Web site where the application was built. RuleGroup, Rule, Variable, and Value inherit the source slot from the upper and related
class. If a class was used in two or more sites, it can have multiple values in the source slot.

In Fig. 5, Delivery is a domain name which is the highest category and “Shipping Rates” is the one of applications which belongs to the Delivery domain. And “Shipping Rates Domestic” is the one of the rule groups of the “Shipping Rates” application.

4.2.2. Representation of variables and values

We define Variable and Value classes to represent variables and values in the ontology as shown in Fig. 4. We define the values relation for the Variable class to connect variables and values.

In the example of Fig. 5, Shipping Method is a variable and has Standard Shipping for its values slot. If Shipping Method has another value, e.g., Two-Day Shipping, we can add it to the values slot and make a new Value instance, Two-Day Shipping.

4.2.3. Representation of rules structure

We define Rule and RuleGroup classes because rules and rule groups are entities as well as variables and values. The important feature of the Rule class is that it generalizes several similar rules in the rule base. We add the rules relation for the RuleGroup class to connect Rule and RuleGroup. In Fig. 5, the rule group Shipping Rates Domestic has the rule Standard Shipping Rule.

IF and THEN represent relationships between rules and variables. Therefore, we define IF and THEN as slots of the Rule class, and variables and values as facets of the IF and THEN slots. For example, the rule Standard Shipping Rule has Shipping Region and Shipping Method for variables, and Domestic and “Standard Shipping” for values in its IF part as shown in Fig. 5.

We excluded connectives like AND and OR from OntoRule because it is hard to represent the complex nested structure of connectives in simple frame representation, and representing all connectives in the ontology has no effect of generalization.

4.2.4. Ontology for the characteristics of variables and values

We described the basic information that we can elicit from rules without additional analysis. However, we need to analyze the information to provide enhanced identification support, and thus we define new characteristics of variables and values.

For the characteristics of variables, we define variableType, unit, range, and omittedCount for slots of the Variable class as shown in Fig. 4. The variableType slot can have one of three types—OAV, Numeric, and Fact. The OAV (Object Attribute Value) type variable can have predetermined string values. In other words, we can select a value for an OAV type variable from predefined strings. The Fact type

IF and THEN represent relationships between rules and variables. Therefore, we define IF and THEN as slots of the Rule class, and variables and values as facets of the IF and THEN slots. For example, the rule Standard Shipping Rule has Shipping Region and Shipping Method for variables, and Domestic and “Standard Shipping” for values in its IF part as shown in Fig. 5.

We excluded connectives like AND and OR from OntoRule because it is hard to represent the complex nested structure of connectives in simple frame representation, and representing all connectives in the ontology has no effect of generalization.
variable is a special case of the OAV type variable. It can have a value Yes or No. The Numeric type variable includes all number type variables such as integer and float. Only the Numeric type variable has the properties unit and range. We represented the patterns of values which the Numeric type variable can have through the unit and range slots. For example, the variable Per Shipment in Fig. 5 is a Numeric type variable. The value of Per Shipment is $3.00 which is composed of unit $ and a number 3.00. The slot, range consists of MIN and MAX facets. In Fig. 5, 1.99 is the MIN value and 31.99 is the MAX value of Per Shipment in the rules of Amazon.com. The slot omittedCount is the number of omission in the rules.

4.2.5. Ontology for synonyms

We define the Synonym class in OntoRule to represent synonyms. An instance of Synonym is connected to an instance of Variable and Value with the synonyms slot. The following example shows a value International Surface and its synonym Standard Surface Mail:

\{
  "International Surface"
  IS-A: Value
  synonyms: "Standard Surface Mail"
\}

\{
  "Standard Surface Mail"
  IS-A: Synonym
  source: Amazon.com
\}

4.3. Ontology generation from rules

In ontology generation and management, there are many issues, such as addition, conflict, and generalization, but we are not discussing these issues and the detailed ontology generation procedure in this paper in order to focus on the design of OntoRule.

Most parts of OntoRule can be automatically generated from the RIML and RSML files acquired from another site. Table 1 shows the sources of ontology and generated parts. Information in OntoRule is similar to the contents of RIML and RSML files, but OntoRule has a higher level of abstraction. The ontology generation procedure combines ontology components from three sources in Table 1.

5. Ontology-based rule identification procedure

In this section, we propose a procedure which automatically identifies rule components by using OntoRule. The procedure is based on four issues described in Section 3.3. First, the procedure selects a proper ontology range. Second, it follows a bottom–up approach in the procedure. Third, it proceeds from variable to value. Fourth, it includes identification of omitted variables. The overall procedure can be summarized as follows:

(1) Determine the text range for a rule group.
(2) Select the ontology range which will be applied to rule identification.

Table 1

<table>
<thead>
<tr>
<th>Source</th>
<th>Generated ontology components</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIML</td>
<td>Variable: Omittedcount</td>
</tr>
<tr>
<td></td>
<td>Synonym: Name</td>
</tr>
<tr>
<td>RSML</td>
<td>RuleGroup: Name, RuleGroup–rule relation</td>
</tr>
<tr>
<td></td>
<td>Rule: Name, IF (variables, values), THEN (variables, values)</td>
</tr>
<tr>
<td></td>
<td>Variable: Name, varaibleType, unit, range (MAX, MIN), Variable–value relation</td>
</tr>
<tr>
<td></td>
<td>Value: Name</td>
</tr>
</tbody>
</table>

The knowledge engineer

<table>
<thead>
<tr>
<th>Domain</th>
<th>Application: Name, Application–RuleGroup relation</th>
</tr>
</thead>
</table>

(3) Identify variables through OntoRule.
(4) Identify values linked to the identified variables through OntoRule.
(5) Identify omitted variables and linked values through OntoRule.
(6) Manually determine the text range for a rule, and search for a similar rule from OntoRule.
(7) Identify IF and THEN parts in the rule by using ontology. Steps 6 and 7 are repeated for all rules in the rule group.
(8) Manually identify remaining variables, values, rules, IF, THEN, and connectives.

We try to describe the rule identification procedure through formal notations to make it easier to understand. We define the notation for the text and ontology as follows:

\[ r: \text{An index of the rule group.} \]
\[ T: \text{The universe of text in the Web page.} \]
\[ T = \{ T_1, \ldots, T_q \}. \]
\[ T_r: \text{The text which implies a rule group, } G_r. \]
\[ O_r: \text{OntoRule which consists of the ontology of rule groups.} \]

5.1. Select \( T_r \)

The knowledge engineer selects a part of the text \( T_r \) from the whole text \( T \) in the Web page. \( T_r \) is the target range for which we want to identify a rule group. Selecting a proper range for a rule group greatly affects rule acquisition performance. An ontology may help to select \( T_r \) by comparing words between the Web page and the ontology. However, we did not include this in the current approach. Instead of providing automated help, we suggested some guidelines for the knowledge engineer based on our experience. First, most tables are good for rules. All rules for shipping rates in BarnesAndNoble.com and Powells.-com are presented in tables. Second, itemized text such as

Please cite this article as: Park, S., Lee, J.K., Rule identification using ontology while acquiring rules from Web pages, International Journal of Human Computer Studies (2007), doi:10.1016/j.ijhcs.2007.02.004
shown in Fig. 2 is good for acquiring rules because it has a structured format.

5.2. Recall O_r

The knowledge engineer manually designates a proper rule group O_r for T_r from the ontology rule group list O, which is shown by OntoXRML. OntoXRML loads O_r and uses it for the rest of the procedure.

5.3. Search for variables from the text

To formally describe the variable and value identification algorithm, we define the notation for variable and value candidates of the text as follows:

\( i: \) An index of variable candidates.
\( j: \) An index of value candidates for a selected variable.
\( V = \{ V_1, V_2, \ldots, V_n \}: \) A set of variable candidates for a selected range of the Web page. \( V_i = \{ \text{variableType}, \text{values}, \text{unit}, \text{rangeMIN}, \text{rangeMAX} \}: \) The knowledge engineer designates a proper rule group O_r for T_r from the ontology rule group list O, which is shown by OntoXRML. OntoXRML loads O_r and uses it for the rest of the procedure.

5.3.1. Search for variables from T_r by using the ontology, O_r

- \( \text{Variable}(O_r) \cap \text{Word}(T_r) \Rightarrow V \).

The words extracted from T_r are represented as Word(T_r). By comparing the words with the variables of O_r, denoted as Variable(O_r), we can elicit V which is an extended set of variable candidates. Variables can be duplicated in V.

5.3.2. Select variables from V

We can calculate the likelihood of variable candidates and prune inappropriate candidates according to the likelihood value. However, in this study, we just select all candidates without pruning them, leaving the pruning for future research. In this process, wrong candidates may be selected for variables, so we can measure the performance of automatic selection in an experiment.

5.4. Search for linked values to V_i from T_r

- For each \( V_i \),
  - If Type\( (V_i) = \text{OAV} \) then \( \text{Values}(V_i) \cap \text{Word}(T_r) \Rightarrow U_i \).

The elicited value candidates for \( V_i \) are represented as \( U_i = \{ u_{i1}, u_{i2}, \ldots, u_{in} \} \). The \( u_{ij} \) element is an OAV type value candidate which is a string. Values can be duplicated in the set as well as variables.

5.5. Search for omitted variables

Searching for omitted variables is the most difficult job if there is no help from the ontology. OntoXRML can automatically find omitted variable candidates and linked values. First, OntoXRML searches for the plausible omitted variables from the ontology. If a variable in the ontology is omitted in the previous rules, it is possible to be omitted also in the current Web pages. Therefore, if omittedCount of a variable is larger than 0, OntoXRML lists the variable in omitted variable candidates. Second, OntoXRML searches for linked values of the variable candidates from the text. If any linked value exists, it means that the variable is omitted in the Web page. Therefore, OntoXRML identifies the variable and linked values.

5.6. Search for a similar rule in the ontology

Once the terms are identified, we need to integrate them into rules, and IF and THEN parts. The procedure consists of two steps: finding a similar rule in the ontology for the given range of the text and finding ranges for IF and THEN in the rule. The two steps are repeated for all rules in T_r.

To formally describe the algorithm, we define the following notation:

\( l: \) An index of rules in a rule group \( G_r \).
\( G_r = \{ R_{r1}, R_{r2}, \ldots, R_{rk} \}: \) The selected rule group in the ontology. \( R_{ri} \) is the \( i \)-th rule of \( G_r \).
\( O_r: \) An ontology for a rule \( R_{ri} \).
\( T_R: \) The text of the Web page which implies a rule. \( T_R \cap T_r \) is a target range in which s/he wants to identify a rule.

5.6.1. Select \( T_R \)

The knowledge engineer designates \( T_R \) to find a similar rule from \( O_r \). \( T_R \) is a target range in which s/he wants to identify a rule.
5.6.2. Calculate similarity of each rule in \(O_r\) for \(T_R\)

- For each \(O_{rl}\) in \(O_r\),

Calculate the following values:

- \(MVU(O_{rl}|T_R)\): The number of variables and values in \(O_{rl}\) matched to the identified variables and values in \(T_R\).
- \(MVU(T_R|O_{rl})\): The number of identified variables and values in \(T_R\) matched to the variables and values in \(O_{rl}\).
- \(VU(O_{rl})\): The number of variables and values in \(O_{rl}\).
- \(VU(T_R)\): The number of identified variables and values in \(T_R\).

\[
\text{Sim}(O_{rl}, T_R) = \frac{MVU(O_{rl}|T_R)}{VU(O_{rl})} \times \frac{MVU(T_R|O_{rl})}{VU(T_R)}.
\]

\(MVU(O_{rl}|T_R)/VU(O_{rl})\) means the ratio of matching variables and values to all variables and values of \(O_{rl}\); while \(MVU(T_R|O_{rl})/VU(T_R)\) is the ratio of matching variables and values to all identified variables and values of \(T_R\). We multiply both of them to make an appropriate measure. If \(O_{rl}\) and identified terms in \(T_R\) exactly match, \(\text{Sim}(O_{rl}, T_R)\) will be 1.

To help readers understand our measure, we demonstrate an example. Fig. 6 shows the selected part of the variable and the value identification result for a rule in Barnes&Noble.com on the returns policies.

OntoXRML identified 4 variables and 7 values. To search for the most similar rule from the ontology, we calculate \(\text{Sim}(O_{rl}, T_R)\) for 2 rules in the ontology as shown in Fig. 7.

In \(\text{Rule 1}\), two variables which are in boldface are matched to variables in the given text. Therefore, \(MVU(O_{Rule 1}|T_R)\) is 2 and \(VU(O_{Rule 1})\) is 4. And, \(MVU(T_R|O_{Rule 1})\) is 2 and \(VU(T_R)\) is 11 since 4 variables and 7 values exist in the text. The following shows the calculation result:

\[
\text{Sim}(O_{Rule 1}, T_R) = (2/4) \times (2/11) = 0.1.
\]

We can calculate the similarity of \(\text{Rule 2}\) in the same manner as follows.

\[
\text{Sim}(O_{Rule 2}, T_R) = (10/19) \times (10/11) = 0.48.
\]

Note that \(MVU(O_{Rule 1}|T_R)/VU(O_{Rule 1})\) and \(MVU(O_{Rule 2}|T_R)/VU(O_{Rule 2})\) are similar as 2/4 and 10/19, respectively. However, \(MVU(T_R|O_{Rule 1})/VU(T_R)\) and \(MVU(T_R|O_{Rule 2})/VU(T_R)\) are very different. That is the reason why we consider not only \(MVU(O_{rl}|T_R)/VU(O_{rl})\), but also \(MVU(T_R|O_{rl})/VU(T_R)\).

5.6.3. Select a rule for \(T_R\)

From the similar rules found, OntoXRML selects only one rule which has the highest similarity. If there are multiple rules with the same similarity, the rule with the largest \(VU(O_{rl})\) is selected among them.

5.7. Assign IF or THEN to identified variables and values

Assigning \(IF\) or \(THEN\) is very simple. If an identified variable belongs to an \(IF\) part of the selected rule in Section 5.6.3, we can assign \(IF\) to the variable. It is same with...
values. We cannot assign IF or THEN to variables and values which are not in the selected rule of ontology.

5.8. Identify remaining rule components manually

The knowledge engineer manually confirms and corrects automatically identified terms and identifies remaining terms which are not recommended through OntoRule. Finally, the knowledge engineer completes rule identification by identifying new rules, IF or THEN if some terms remain unassigned to any rule, IF, or THEN. Also, s/he identifies connectives such as AND, OR.

6. Performance of using ontology in rule identification

6.1. Experiment design for evaluation

In this section, we describe evaluating the performance of our approach proposed in Section 5, that is, the effect of automatic rule identification using OntoRule. In order to start the experiment, we need an ontology to be applied in rule identification. Therefore, we first constructed the ontology from Amazon.com. The ontology consists of 1 domain, 1 application, 13 rule groups, 36 rules, 17 variables, and 494 values.

For the evaluation, we selected two well known online bookstores: Barnes&Noble.com (in short BN), and Powells.com (in short Powells). From these sites, we analyzed the Web pages on the shipping rates, free shipping, and returns policies, and identified rules to show the performance of using OntoRule in rule identification. We conducted the experiment in the order of BN and Powells. And, we modified the ontology after rule acquisition from BN to test the ontology learning effect in Powells.

In the performance evaluation, we have the following concerns:

1. How many variables and values can be automatically identified through OntoRule?
2. How many variables and values are wrongly recommended from OntoRule?
3. Is there an ontology learning effect?
4. How many omitted variables and values can be automatically identified through OntoRule?

To measure the above concerns, we adopted the standard measures for information retrieval (IR), which are recall and precision.

We define the metrics as follows:

\[ CR = \frac{Precision_{Total}}{Recall_{Total}} \]

(1)

\[ Recall = CR / TT \]

(2)

We calculated the above metrics in terms of variable and value. We represented Precision and Recall on variable and value, respectively—Precision(V), Precision(U), Recall(V), and Recall(U).

6.2. An example of the experiment on text

Fig. 8 shows an acquired rule from the text in Fig. 2. Correctly recommended variables and values are in boldface. Therefore, we can calculate measures as follows:

\[ CR(U) = 8, \quad DR(U) = 8, \quad TT(U) = 13, \]

\[ Precision_{Total} = 17/17 = 100.0\%, \]

\[ Recall_{Total} = 17/26 = 65.4\%. \]

Compared to the result of the identification without learning in Table 4, Precision and Recall are relatively high. It means that the rule in Fig. 8 is a good case in our experiment. Normally, rules in a table have higher performance than rules in text because they are more structured and typical.

6.3. Experiment results

The overall statistics of existing rule components in BN and Powells are summarized in Table 2. We did not include Amazon in our experiment because we acquired the ontology from Amazon, which means that the evaluation result of Recall in Amazon will be 100%. BN has 11 rule groups on shipping rates of each geographic region and returns policies. Powells has 6 rule groups on shipping rates, free shipping, and returns policies.

Table 3 shows the contribution of using ontology in variable and value identification. Precision(V) and Recall(V) are 72.96% and 82.96%, respectively. Precision(U) and Recall(U) are high at 97.45% and 94.16%. Especially, Precision(U) and Recall(U) of Powells are very high at 99.65% and 99.11%. This happens because Powells uses large tables for shipping rates which are well structured and easy to match with the ontology.

We obtained satisfactory performance of using ontology in rule identification from our experiment. Precision\(_{total}\) and Recall\(_{total}\) are 93.89% and 92.74%, respectively. It means that 93.89% of automatically identified terms are correct and 92.74% of all existing terms are correctly identified from the ontology. Precision and Recall are plausible even though it is hard to compare with other research on the performance.

To test the ontology learning effect, we conducted two separated rule identification experiments from texts of

Please cite this article as: Park, S., Lee, J.K., Rule identification using ontology while acquiring rules from Web pages, International Journal of Human Computer Studies (2007), doi:10.1016/j.ijhcs.2007.02.004
Powells. One used original Amazon ontology and the other used modified Amazon ontology with rules of BN. Table 4 shows the difference between them. The experiment shows that the ontology learning effect exists. We expected that both CR and DR would increase as the ontology grows. In the experiment, CR increased from 23 to 28, and DR increased from 28 to 34 as we expected. As the result of the ontology learning effect, both Precision and Recall increased.

Table 5 shows the contribution of using ontology in omitted variable identification. Precision$(V)$ and Recall$(V)$ for omitted variables are 84.49% and 85.87%, respectively. Besides Precision$(V)$ and Recall$(V)$, note that 158 omitted variables were automatically identified out of 270 identified variables. It shows that automatic identification of omitted variables is important in rule identification.

6.4. Discussions on the experiment

In the experiment, we accepted all recommendations from ontology without pruning. However, it is possible to enhance Precision by pruning the recommendations especially from the text. Developing a pruning strategy and algorithm will be our next research issue. Also, the experiments are not sufficient for testing all features of our procedure such as rule, IF, and THEN identification. We are planning to extend experiments for those features and a pruning strategy.

One limitation of the evaluation is that there are only two sites in our experiment. The objective in this experiment is not to empirically verify the validity of our approach, but to show an example where our approach works well. However, an extended experiment with enough Web sites is surely required to verify and generalize our approach, so we are planning to do that.

7. Conclusion

To reduce the knowledge engineer’s manual work in rule identification of the XRML approach, we proposed the ontology-based methodology of enhanced rule identification. As the first part of the methodology, we have designed an ontology, OntoRule which includes information on rule components and structures for automated rule identification. While designing OntoRule, we considered the features concerning the ontology learning effect, a bottom–up approach of rule identification, management of the ontology range, detection order of variable and value, and detection of omitted variables. Also, we proposed a detailed procedure of rule identification using OntoRule.
The procedure is based on the same issues as the design of OntoRule.

In the experiment, the enhanced XRML editor, OntoXRML identified 92.74% of all existing variables and values in BN and Powells Web pages and correctly identified 93.89% of all recommended variables and values. The results are plausible and interesting even though it is difficult to compare these results to the results of other approaches. We expect that our research shows the potential and value of using ontology in rule acquisition and deals with various issues on it.

There are some future research issues to enhance the proposed approach. First, we can exploit some pruning strategies for choosing variables, values, and rules in the algorithm. The new experiment should include performance evaluation on the strategies. Second, identifying rules from identified variables and values can be improved. We are planning to fully automate rule identification by incorporating it into variable and value identification. Last, we need to generalize our methodology in various domains to verify the effect.

Acknowledgement

This research is partially funded by the Research Grant (C220/MSS5C001) of Singapore Management University.

References


Liegib, T., Noppens, O., 2004. ONTOTRACK: combining browsing and editing with reasoning and explaining for OWL lite ontologies. Lecture Notes in Computer Science 2870, 244–258.


