SEMI-AUTONOMOUS RULE ACQUISITION FRAMEWORK USING CONTROLLED LANGUAGE AND ONTOLOGY

Mye M. Sohn and Yungyu Choi
Department of Systems Management Engineering, Sungkyunkwan University
300, Chunchun-dong, Suwon, Kyunggi-do, 440-746, Korea

Keywords: Rule extraction, Controlled language set, Ontology, Rule Markup Language, XRML.

Abstract: This paper presents a framework for rule extraction from unstructured web documents. To do so, we adopted the controlled language technique to reduce the burden as well as error of a domain expert and suggest a rule extraction framework that uses ontology, to solve the problem of missing variable and value that may be caused by incomplete natural language. Here, it is referred to as NEXUCE (New rule EXtraction Using ontology and Controlled natural languagE). To evaluate the performance of the NEXUCE framework, the natural language statements were collected from the websites of Internet bookstores and the rule extraction capability was analyzed. As a result, it was proven that NEXUCE can have more than 70% of rule extraction from unstructured web documents.

1 INTRODUCTION

There has been a great deal of research in the field of rule extraction from web documents to provide advanced intelligent service in semantic web era. The technique for these can be categorized into: web mining, natural language processing or controlled natural language, diagrammatic approach and markup language. Lately, a great attention has been shown in applying ontological techniques to support rule extraction from web documents (Vargas-Vera et al, 2001, Cimiano and Handschuh, 2002, Alani, et al, 2003, and Park and Lee, 2007). Ontoloies have been used to sharing and reuse domain-specific vocabularies and their relationships that can be adopted to generation of common understanding rule. However, rule extraction is a still difficult task to all the knowledge engineers and domain experts even though various tools and methodologies are proposed. Because knowledge engineers do not have sufficient knowledge about domain of discourse and domain experts are ignorant to rule extraction methodologies and technologies.

To alleviate this difficulty, we propose a rule extraction methodology, named controlled natural language and ontology. New rule EXtraction Using ontology and Controlled natural languagE (NEXUCE, it is pronounced as nexus) that domain experts can superintend the overall rule extraction procedure. A controlled natural language is a subset of natural language that is obtained by restricting grammars and vocabularies to reduce or eliminate ambiguity in natural language (Schwitter and Tilbrook, 2004, and Thomson and Pazandak, 2005). Recently, some researchers argued that it can be knowledge sharing between human and machines (Schwitter and Tilbrook, 2004). By adopting controlled natural language, burdens of a domain expert caused by learning of rule acquisition method, language and tool can be reduced to some extent. An ontological approach can be used to define the vocabularies and their relationship to achieve a common understanding about domain of discourse. In this paper, ontology applies to generate the structured document which is implied primitive statements (such as IF and THEN), connectives (such as AND, OR, and NOT), and operators (such as GT, GE, LT, and LE). Also, ontology is able to be adaptively refined according to newly acquired rules in the domain of discourse.

This paper is organized as followed. Chapter 2 presents reviews of the related researches and addresses their limitations. In Chapter 3, we first present overall architecture of the proposed system. In Chapter 4, we present an ontology refinement procedure. Then we implement and demonstrate the NEXUCE prototype in Chapter 5. Also, we show the

Sohn M. and Choi Y. (2009).
SEMI-AUTONOMOUS RULE ACQUISITION FRAMEWORK USING CONTROLLED LANGUAGE AND ONTOLOGY.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 238-245
DOI: 10.5220/0001658702380245

performance of our system. Finally, we summarize our research contribution with some concluding remarks in Chapter 6.

2 RELATED LITERATURE

2.1 Rule Extraction

We can classify existing methodologies to extract rules from web documents into five categories: natural language processing, text mining, diagrammatic approach, rule markup language and ontologies. Table 1 shows those three categories, summarized technical features of category and applied technologies or standards.

Table 1: Categories of rule extraction methods.

Type of method	Technical features	Technologies, methods or standards
natural language processing or computational linguistics	Rule is derived from speech and language processing such as parsing and tagging (Gelbukh, 2005)	Model-based processing (state machines, rule systems, logic, probabilistic models, and vector-space models), search, and machine learning, etc
Text mining	Rule is generated by the discovering of patterns and trends in web document (Etchells and Lisboa, 2006 and Ressom, et al., 2006)	machine learning techniques such as inductive learning, neural networks, and statistical models, statistical pattern learning and statistics, etc
Diagrammatic approach	Rule is extracted through graphical rule representation interface	Conceptual graph, decision table, and influence graph
Rule markup language	Rule is identified and expressed with the annotation tags	XML, RDF(S), OWL,
Ontological Approach	Rule is derived through the defining of vocabularies and their relationships to achieve common understanding about domain of discourse	XML, RDF(S), OWL, and reasoning, etc

Let us examine the pros and cons of each method motioned above. The most comprehensive method is the natural language processing (NLP) or computational linguistics. The goal of the NLP is to develop procedures which make it possible to process the informational contents of texts and conversation, learn more about language structure (Kent A., et al., 1975), and share the informational contents between human and machine. Several algorithms, tools, and implementations for NLP have been proposed (Bernstein, et al., 2005; Bernstein, et al., 2006; Wang, et al., 2007; and Thompson, et al., 2005). However, NLP has some shortcomings in its abilities to identify the role of a noun phrase, represent abstract concepts, classify synonyms, and represent the sheer number of concepts needed to cover the domain of discourse (Sullivan D., 2001). Text mining is defined as the discovery of previously unknown knowledge in a text. It is a subfield of NLP, and inherits a set of fundamental analysis tools from NLP.

2.2 Controlled Language Set

NEXUCE is also related to the studies on the controlled natural language processing with menu-based interface which was proposed as a subset of natural language processing. As mentioned earlier, a controlled natural language is obtained by restricting grammars and vocabularies to reduce or eliminate the ambiguity in the natural language. Recently, some researchers argued that it can be a knowledge sharing between human and machines (Schwitter and Tilbrook, 2004). To promote knowledge sharing between human and machines, some researches which are called the menu-based natural language interface are performed in the area of command and query generation or search engine. LingoLogic as a menu-based natural language interface (MBNLI) system restricts the user from performing commands and queries that underlying systems can understand (Thompson, et al., 2005). Ginseng is a search engine with an induction method to convert the natural language into RDQL (RDF Data Query Language), a query language for semantic web (Bernstein, et al., 2005), and GINO, which utilizes the controlled language set technique based on system induction in order to add the class and attribute of ontology (Bernstein, et al., 2007). PANTO converts the query prepared by natural language into RDQL and queries RDF (Wang, et al., 2007). NEXUCE focuses on devising a new rule acquisition mechanism for web documents by utilizing the controlled natural language processing with the menu-based interface, same as former researches. However, by adopting ontological technology and applying to the rule extraction, the application spectrum natural language processing with the menu-based interface is widened to some extent.

2.3 Extensible Rule Markup Language

Since XRML, a rule markup language that can identify and structure the implicit rules embedded in Web pages, was suggested by Lee and Sohn (Lee and Sohn, 2003), follow-up researches have been performed. Kang and Lee proposed XRML 2.0 as a revised edition of XRML 1.0 (Kang and Lee, 2005). It expands reserved words of XRML 1.0 and adds new operators to identify and generate a structured rule. OntoRule, another version of XRML was proposed by Park and Lee. It adapted the rule ontology which is acquired from rule bases of a similar domain as a rule acquisition tool (Park and Lee, 2007). However, it is still a difficult task for a domain expert who has a great store of domain knowledge but is ignorant to XRML syntax and tool to extract the rule from web documents even though we applied XRML. To overcome the limitation of the XRML, we tried to expand the XRML with the aim of rule acquisition by the domain expert who does not have any skill or knowledge about rule acquisition. The domain expert composes a rule either by typing it in or selecting items from a series of menu-based rule extraction interface.

In this paper, the NEXUCE was developed to support the full procedure of the rule extraction by using a controlled language set and ontology. Using the NEXUCE editor may prevent the failure of a rule generation likely resulting from underestimating or overestimating the capability of a knowledge engineer (Thompson, et al., 2005). Chapter 3 will describe the architecture of the NEXUCE and the rule extraction procedure using the architecture.

3 OVERALL ARCHITECTURE OF NEXUCE

NEXUCE, a new framework for the rule extraction implicitly contained in the web document, is consisted of four parts such as Controlled Rule Language Interface, Rule-based Variable and Value Identification Module, Ontology-based Rule component Identification Module, and Structured statement Generation Module. The Controlled Rule Language Interface receives natural-language statements from the domain expert and generates a structured statement step by step through graphic user interface (GUI). Figure 1 illustrates the overall working procedure of NEXUCE.

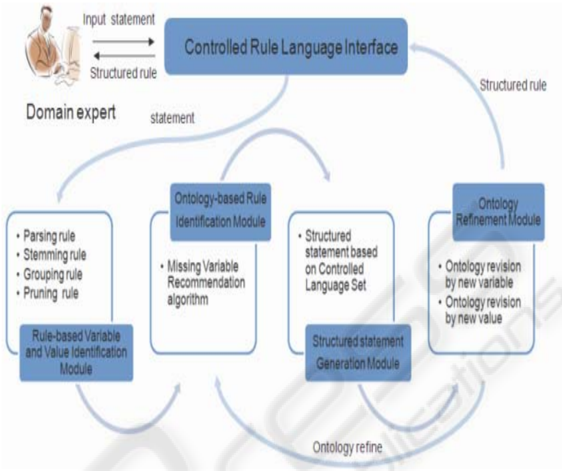

Figure 1: Overall working procedure of NEXUCE.

In this paper, we define a structured statement as a natural-language statement that is primitive statements (such as IF and THEN), connectives (such as AND, OR, and NOT), and operators (such as GT, GE, LT, and LE). For instance, if Controlled Rule Language Interface receives a natural language statement '`We can ship to an address in Korea`,' then NEXUCE returns a structured statement '`Delivery policy is that if country is Korea, then delivery is allowed`' through Controlled Rule Language Interface. Key points of converting a natural language statement into a structured statement are exact parsing and regrouping of parsed words to suit the rule structure.

3.1 Rule-based Variable and Value Identification Module

To extract IF-THEN type rule which is implied in natural-language statements, variables and values of IF and THEN parts should be identified. The major function of Rule-based Variable and Value Identification Module is to analyze the natural-language statement, group the parsed words to suit the rule structure, and identify the components like variables and values of rules. The natural-language statement that we will deal with is restricted only to the statement because of the restriction of the parser. Rule-based Variable and Value Identification Module adapts Stanford parser that can parse 90% or more of the natural-language statement to get word components and their part of speech (Wang, et al., 2007 and Klein and Manning, 2003). For instance, the following statement shows a part of the document relating to delivery policy that amazon.com published on their web site.

> We are currently able to ship books, CDs, DVDs, VHS videos, music cassettes, and vinyl records to European addresses. ***We can also ship some software, electronics accessories, kitchen and housewares, and tools to addresses in Denmark, Finland, France, Germany, Ireland, the Netherlands, Sweden, and the United Kingdom.***

If we apply Stanford parser to analysis the above first statement, we get the rooted spanning tree. The root of the tree is statement (S) and it has three branches such as a noun phrase (NP), a verbal phrase (VP) and the full stop (.). The parsed statement is regrouped according to the following rules.

- Rule 1: generation and stemming of parsed word set

To identify the word set from rooted spanning tree which is generated by Stanford parser, we adopt the depth-limited search (DLS). Depth-limited search traverses the rooted spanning tree until no more sub-NP nodes exist. The output of DLS is a set of parsed words, $PW = \{PW_1, PW_2, \ldots, PW_n\}$ where PW is a parsed word which has the lowest NP node as a super node. At the moment, the plural is replaced with the singular.

- Rule 2: generation of coined word

If the two or more words share the lowest NP node, these words treat a word and insert '_' as a connective. If VHS and video share a super-NP node, we treat two words as a word. As a result, a coined word such as 'VHS_video' is generated, parsed word set is modified as follows: $PW = \{PW_1, PW_2, \ldots, PW_m\}$ where $n \geq m$.

- Rule 3: grouping for parsed word set

To group the parsed word set, we return to rooted spanning tree of the structured statement. If arbitrary two words in parsed word set are not adjacent, two words are grouped as a different word group. Adjacent node means two nodes that do not hold any word except 'and' or 'or' between two nodes. Group of parsed word set (GPW) is generated after rule 3 is been applied.

- Rule 4: pruning for GPW

As a final step, words which has pronoun are deleted from grouped parsed word set. It is called pruned group of parsed word (PGPW) set.

The domain expert performs a refinement to PGPW because Rule-based Variable and Value Identification Module can't perfectly group all kinds of natural-language statement. The refined PGPW is then delivered to the Ontology-based Rule component Identification Module to identify the components of a complete rule.

3.2 Ontology-based Rule Component Identification Module

The pruned group of parsed word set may be used to the variable and/or value of IF-THEN rule. Ontology-based Rule component Identification Module takes an arbitrary set of PGPW and forms it into IF-THEN rule. To achieve this, this module has to determine a pair of variable-values of IF and THEN part of rule. However, also, it has to identify or recommend the missing variable of if-then rule to the domain expert because whole components for rule forming may not identified by Rule-based Variable and Value Identification Module due to the incompleteness of a natural-language statement. To do so, we adapt an ontology that can model the concept (e.g., variable and value) and relationship of concepts and provide a shared and common understanding of the domain that can be communicated between human and machines (Davies et al, 2002).

We use an ontology called $NEXUCE_{Ont}$ which can be used to identify the missing variable of a rule to be generated. Also, ontology matches variables and values induced by Rule-based Variable and Value Identification Module. In the above example, the domain expert has only imperfect rule components such as {*software*, *electronics_accessory*, *kitchen*, *houseware*, *tool*}, {*address*}, and {*Denmark*, *Finland*, *France*, *Germany*, *Ireland*, *the_Netherlands*, *Sweden*, *the_United_Kingdom*}. The domain expert exactly doesn't know what variables are adequate for the missing variable of the rule although s/he has an idea that the imperfect rule components may contain some rules. To support the domain expert, we propose Missing Variable Recommendation (MVR) algorithm to recommend a set of missing variable. MVR algorithm is summarized in Figure 2.

```
Function returnMissingVariables
(words, ontology)

create a tree according to the
hierarchy of the ontology
find words in the tree and set the
nodes as wordNode[]

  for h ← 1 to
wordNode[].totalNumber
   currentNode ← wordNode[h]
   while currentNode != rootNode
     currentNode.count ++
     currentNode ←wordNode
      [h].parentNode

  for I ← 1 to
wordNode[].totalNumber
    currentNode← wordNode[i]
   while currentNode.parentNode !=
         rootNode
      if currentNode.count !=
         currentNode.parentNode.coun
         t
          & currentNode.parentNode
         doesn't exist in variable[]
        save currentNode.parentNode
        in variable []
        currentNode ←
        wordNode[i].parentNode
```

Figure 2: Missing Variable Recommendation (MVR) algorithm.

The MVR algorithm adapts the reversed depth-first search to get the missing variable from the ontology. The input of MVR algorithm is all elements of identified rule components, and output is a set of concepts which is mapped to the variable of the rule.

3.3 Structured statement Generation Module

Structured statement Generation Module converts the proposed set of the variable and value induced by MVR algorithm to the structured statement. In this stage, the domain expert can generate a structured statement. In this paper, we propose syntaxes of a structured statement based on controlled language set which allows the restricted vocabulary set with a single defined meaning and controlled grammar usage. This module generates a set of structured statement as depicted Figure 3.

The domain expert only determines and selects a structured statement that suits its purpose. Finally, the selected structured statement is converted into the canonical rule that is appropriate to inference engine. Controlled Rule Language Interface can support overall procedures of the rule extraction.

No	variable	values
1	address	Denmark, Finland, France, Germany, Ireland, Netherlands, Sweden, United Kingdom
	item	software, electronics accessory, kitchen and housewares, tool
2	address	Denmark, Finland, France, Germany, Ireland, Netherlands, Sweden, and United Kingdom.
	item	kitchen and housewares, tool
	consumer electronics	software, electronics accessory
3	address	Denmark, Finland, France, Germany, Ireland, Netherlands, Sweden, and United Kingdom.
	item	kitchen and housewares, tool
	consumer electronics	electronics accessory
	computer product	software

Structured NL statement
If (address) **is** (Denmark, Finland, France, Germany, Ireland, Netherlands, Sweden, United Kingdom) **and** (item) **is** (software, electronics accessory, kitchen and housewares, tool), **then** ~
If (address) **is** (Denmark, Finland, France, Germany, Ireland, Netherlands, Sweden, United Kingdom) **and** (item) **is** (kitchen and housewares, tool) **or** (consumer electronics) **is** (software, electronics accessory), **then** ~
If (address) **is** (Denmark, Finland, France, Germany, Ireland, Netherlands, Sweden, United Kingdom) **and** (item) **is** (kitchen and housewares, tool) **or** (consumer electronics) **is** (electronics accessory) **or** (computer product) **is** (software), **then** ~

Figure 3: Generated structured statements by structured statement Generation Module.

4 ONTOLOGY REFINEMENT PROCEDURE

$NEXUCE_{Ont}$ models concepts, their relationship and instances in the domain of discourse. Also, it specifies class hierarchy, synonym, and/or equivalent relationship between classes. However, the ontology development is still a bottleneck to the knowledge engineer who sufficiently doesn't have domain knowledge even though s/he has genuine ontology editor. One way to cope with this bottleneck of the ontology generation is to refine the ontology continuously. The knowledge engineer develops an initial rough ontology based on his/her incomplete domain knowledge at the initial stage. The rough ontology is continuously refined by the newly generated rule that is reflected in the domain knowledge of domain experts.

To suggest the ontology refinement method, we assume that n_{newvar} and n_{newval} can be associated with concepts in acyclic graph which is induced based on inherited hypernym hierarchy in WordNet. A node n_{newvar} and n_{newval} are the new variable and value which are induced by SGM but may or may not be modeled in $NEXUCE_{ont}$. In this paper, the ontology refinement is progressed by two ways: new value insertion, and new variable insertion.

4.1 New Value Insertion

In the case of new value insertion to $NEXUCE_{ont}$, the newly generated rule by SGM contains new values which are not modelled in $NEXUCE_{ont}$. The basic underlying idea of our method is to refine a

$NEXUCE_{ont}$ by reflecting the domain expert's knowledge that is melted in the newly generated rule. We call this the repetitive ontology refinement approach. This approach is summarized as follows.

The superordinate node of n_{newval} should be identified in order to insert the new value to $NEXUCE_{ont}$. We calculate the conceptual similarity of n_{newval} and a whole subordinate node of n_{newvar} in $NEXUCE_{ont}$. Acyclic graph which is needed to compute the conceptual similarity of two nodes is induced based on inherited hypernym hierarchy in WordNet. The conceptual similarity of two nodes is considered in terms of node distance. The similarity then between the two nodes is approximated by the number of arcs on the least common superordinate node in the inherited hypernym hierarchy in WordNet. As such, the conceptual similarity of two nodes n_1 and n_2 can be expressed as:

$$\mathrm{ConSim}(\mathrm{n}_1, \mathrm{n}_2) = 1 - \log\left[\frac{(N_1 + N_2)}{2}\right]$$

where definitions of N_1 and N_2 are depicted in Figure 4, and log $[(N_1+N_2)/2]$ is the semantic distance of arbitrary two nodes on the hypernym hierarchy graph. If total number of arcs on the path on n_1 to n_2 is greater than 20, we assume that the conceptual similarity between n_1 and n_2 is '0'. After calculating the conceptual similarity measure in whole pairs of n_{newval} and subordinate nodes of n_{newvar} in $NEXUCE_{ont}$, the superordinate node of node which has maximum similarity can be determined as the superordinate node of n_{newval}.

For example, if we take 'CD' and 'entity' as a n_{newval} and n_{newvar}, we wish to discover the superordinate node of DVD on $NEXUCE_{ont}$. To do so, we induce acyclic graphs which are depicted as Figure 4. In this example, 'CD,' 'computer mouse,' and 'software' are subordinate nodes of the entity.

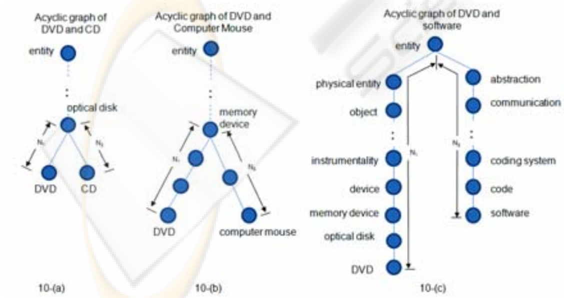

Figure 4: Acyclic graphs of inherited hypernym in WordNet.

The conceptual similarity between two nodes is calculated as below:

Conceptual Similarity (DVD, CD) = 1
Conceptual Similarity (DVD, computer mouse) = 0.602
Conceptual Similarity (DVD, software) = 0.155

The 'optical disk' is recommended as a superordinate node of DVD. As a result, a value is inserted to $NEXUCE_{ont}$. The value insertion is performed continuously whenever a new value is identified from web documents.

4.2 New Variable Insertion

We design the NEXUCE that it can propose an adequate variable for the identified value set. However, the domain expert may want to specify a new variable instead of the variable proposed by the NEXUCE. At this point the newly specified variable by domain expert is added as the subordinate node of the node's superordinate node recommended by the NEXUCE and it relates that variable to the '*owl:equivalentClass*'. Using this relationship '*owl:equivalentClass*', the newly specified variable is also recommended to the domain expert as an alternative when performing the rule extraction in the future.

As mentioned in the statement before, the ontology refinement procedure is performed recursively. The advantage of the recursive ontology refinement process is a two-fold. First, the development burden of the domain ontology which has been generated by a part of an ontological approach for rule extraction will be reduced. Second, this refinement process contributes in extracting the fine rule that precisely reflect implicit domain knowledge. Implicit domain knowledge is defined as a knowledge that has to reflect the rule although it does not represent on natural-language document.

5 EVALUATION

To evaluate the rule extraction capability of NEXUCE, we collect the 125 natural language statements which are posted on amazon.com, barns and noble and etc. Among them, there were 83 statements, and Table 2 shows the results where the NEXUCE was applied to these statements.

In Table 2, S means statement and SS means structured statement. Likewise, $SNLS_s$, $SNLC_{cm}$ and $UNLS_{fail}$ mean the number of structured natural language statements created with no meaning changed, the number of structured NL statements created with meaning changed and the number of the statements that contain the rule but fail to be discriminated.

Table 2: Experiment result.

Site	description	no. of statement/values	$SNLS_s$	$SNLC_{cm}$	$UNLS_{fal}$	Success rate
Amazon.com	DS/SS	27/4	23	1	3	76.2%
	Identified values from unstructured NL statement	167	162	5	0	97.0%
Barnsandnovels. com	DS/SS	15/17	11	1	3	73.3%
	Identified values from unstructured NL statement	65	57	8	0	87.7%
Powells.com	DS/SS	17/0	11	5	1	64.7%
	Identified values from unstructured NL statement	47	42	5	0	89.4%

If using the NEXUCE from the above results, it was found that the rule contained in the descriptive natural language statement is discriminated about 71% on average. In addition, the values contained in the descriptive natural language statement were discriminated about 91.4% on average.

6 CONCLUSIONS AND FURTHER RESEARCHES

In this paper, we proposed a rule extraction framework to support the domain experts who are ignorant to rule extraction methodologies and procedure but have a great store of domain knowledge. A controlled language set and ontology-enabled rule extraction technique is adopted for the framework. The framework includes four parts: Rule-based Variable and Value Identification Module, Ontology-based Rule component Identification Module, Structured statement Generation Module, and Ontology Refinement Module. Also, wee demonstrate the possibility of our controlled language set and ontology-enabled rule extraction framework with an experiment.

Contributions of this study can be summarized as follows. First, we applied rule and graph search technique to formalize structured statement. Second, we devised a new rule extraction framework to support the domain experts. Finally, ontology refinement algorithm is proposed in order to adapt the newly inserted class, e.g. value or variable.

Nevertheless, the study suffers from the limitations that the NEXUCE framework may discriminate only if-then type rules contained in the descriptive statement, the limited ontology was implemented only for the prototype system and various possible exceptions may not be considered and should be researched in future studies. We are planned to evaluate the proposed framework to other rule acquisition approaches.

ACKNOWLEDGEMENTS

This work was supported by grant No. R01-2006-000-10303-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

REFERENCES

Alani, H., Sanghee Kim, Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H. and Shadbolt, N. R., 2003. Automatic ontology-based knowledge extraction from Web documents, *IEEE Intelligent Systems*, Vol.18, No.10.

Bernstein, A., Kaufmann, E. and Kaiser, C., 2005. Querying the Semantic Web with Ginseng: A Guided Input Natural Language Search Engine, *15th Workshop on Information Technology and Systems (WITS 2005)*.

Bernstein, A. and Kaufmann, E., 2006. GINO – A Guided Input Natural Language Ontology Editor, *5th International Semantic Web Conference*, Vol 4273.

Duke, J., Davies, A. and Stonkus, A., 2002. OntoShare: Using Ontologies for Knowledge Sharing, *Proceedings of the WWW Workshop on Semantic Web.*

Etchells, T. A. and Lisboa, P. J. G., 2006. Orthogonal search-based rule extraction (OSRE) for trained neural networks: a practical and efficient approach, *Neural Networks*, Vol.17, No. 2.

Gelbukh, A., 2005. Natural Language Processing, *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS'05).*

Handschuh, S., Staab, S. and Ciravegna, F., 2002. S-CREAM – Semi-automatic CREAtion of Metadata, *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, Vol 2473.

Kang, J. and Lee, J. K., 2005. Rule Identification from Web Pages by the XRML approach, *Decision Support Systems*, Vol. 41, No 1.

Kent, A., Lancour, H., Nasri, W. Z., Daily, J.E., 1975. Encyclopaedia of Library and Information Science, *CRC Press*.

Klein, D. and Manning, C. D., 2003, Accurate Unlexicalized Parsing, *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL '03)*, Vol. 1.

Lee, J.K. and Sohn, M., 2003. Extensible Rule Markup Language – toward Intelligent Web Platform, *Communications of the ACM*, Vol. 46, No 5.

Park Sangun and Lee, J. K., 2007. Rule identification using ontology while acquiring rules from Web pages, *International Journal of Human-Computer Studies,* Vol. 65, No 7.

Ressom, H. W., Varghese, R. S., Orvisky, E., Drake, S. K., Hortin, G. L., Abdel-Hamid, M., Loffredo, C. A.

and Goldman, R., 2006. Biomaker Identification and Rule Extraction from Mass Spectral Serum Profiles, *Computational Intelligence and Bioinformatics and Computational Biology (CIBCB '06).*

Schwitter, R and Tilbrook, M., 2004. Controlled Natural Language meets the Semantic Web, *Australasian Language Technology Workshop 2004 (ALT '04*).

Sullivan, D. (2001) Document Warehousing and Text Mining, Wiley.

Thompson, C. W., Pazandak, P. and Tennant, H. R., 2005. Talk to your Semantic Web, *IEEE Internet Computing*, and Vol. 9, No. 6.

Vargas-Vera, M., Motta, E. and Domingue J.. 2001. Knowledge Extraction by using an Ontology-based Annotation Tool, *K-CAP 2001 workshop on Knowledge Markup and Semantic (K-CAP '01*).

Wang Chong, Miao Xiong, Qi Zhou and Youg Y., 2007. PANTO: A Potable Natural Language Interface to Ontologies, *4th European Semantic Web Conference (ESWC '07),* Vol. 4519.