Ontology Guided Semantic Self Learning Framework

Darshika N. Koggalahewa and Asoka S. Karunananda

***Abstract*—Ontology Guided Semantic Self Learning Framework is a software which is capable of learning from natural language sources. Knowledge extraction process is performed by modeling ontology from the natural language text. The system uses its acquired knowledge to perform teaching and explaining activities to its end users. Self-learning and question based learning modules are two main modules of the system. Semantic learning module captures the knowledge from the text and represents them in an ontology based environment. Question based learning system performs teaching activities to end users. The Natural language learning process achieved seventy percent accuracy while ontology construction process achieved over sixty five percent of accuracy. The research solved the problem of lack of understanding and capturing the knowledge from Natural Language sources and represents the captured knowledge in an automated and efficient manner. It introduced a novel approach of knowledge capturing from unstructured natural language text.**

***Index Terms*—Knowledge, learning, natural language, ontology.**

I. Introduction

Intelligent machines have been introduced to automate humans' activities. Workload towards learning process of a human has not reduced into a considerable extent due to lack of systems which have natural language capabilities. Though the knowledge can be stored inside the computer, (many knowledge representation techniques available), they all suffer from the lack of acquisition capabilities due to various inconsistencies made in representation. Several researches have been made on this particular domain over the past years to achieve a machine with natural language learning capabilities. Most of them have not focused on building a system with both learning and teaching capabilities and also they were not able to ignore the human involvement in extracting the actual meaning of the processed text content. That indicates, still the machine needs more human assistance in learning. According to the investigation, there is no exact way of understanding the natural language robustly, and answer generation by machine itself, from its own knowledge area, which is provided by the domain specific user.

Understanding human intelligence has been a constant preoccupation of scientists. This subject has been considerably developed, offering better understanding about learning and its relationship with knowledge in general, and lately with knowledge in digital form. Getting to know is a part of intelligence, and intelligence cannot be separated from knowledge. As knowledge we designate not only what can be written down in books, but also knowledge gathered through competencies, attitudes, expertise, etc. are also equally important. Knowledge acquisition is important for two basic activities such as learning and teaching. Learning is acquiring new knowledge, behaviours, skills, values or preferences.

It may involve processing different types of information. Learning functions can be performed by different brain learning processes, which depend on the mental capacities of learning subject/agent, the type of knowledge which has to be acquitted, as well as on socio-cognitive and environmental circumstances [1]. Most of the time, human learning depends on reading and understanding a written text. The information available in digital format is growing day by day in millions and trillions of sources. Nevertheless the problem arises when people trying to acquire the knowledge from those contents and sources.

The process is more time consuming and cumbersome. Several efforts have been made to build up a computer system which is capable of acquiring the domain knowledge from the Natural language text. The effort is not yet succeeded due to the ambiguity, complexity, and the different written patterns of the human. The technology is still struggling with implementing a machine which is capable of processing the natural language and acquires the knowledge. Ontology construction for any domain is a labour intensive and complex process. Any methodology that can reduce the cost and increase efficiency has the potential to make a major impact in the life sciences.

The technology of natural language processing has evolved in different areas where some of the efforts have considered on information extraction, Natural language annotation and so on. Though these efforts have achieved in greater success still there is lack of persistence in processing complete natural language. The efforts made on the domain are still in the status of processing controlled natural language instead of complete natural language. Knowledge Representation in computer environment had a significance impact on developing intelligent computer systems. Among the various knowledge representation techniques, ontology is the most powerful and the latest technology of knowledge representation. Nevertheless there is a serious gap between representing the knowledge and constructing the ontology from natural language. The retrieval process of the stored knowledge becomes more significant since it fulfil the basic expectation of the knowledge representation and natural language learning.

Several efforts have been made over the last two decades to

Manuscript received September 1, 2014; revised January 16, 2015. Ontology Guided Semantic Self Learning Framework is a research conducted for the partial fulfilment of the requirements of the Degree of MSc in Artificial Intelligence.

Darshika N. Koggalahewa was with the Universtiy of Moratuwa, Sri Lanka. He is now with Sri Lanka Institute of Information Technology, Sri Lanka (e-mail: darshika.k@sliit.lk).

Asoka S. Karunananda was with the University of Moratuwa. He is now with Sir John Kotelawala Defence University, Sri Lanka (e-mail: .

DOI: 10.7763/IJKE.2015.V1.5

build software capable of learning from natural language sources, understanding and self- explaining. Mohamed Yehia Daha, Hesham A. Hassan and Ahmed Rafea [2] have developed an approach for constructing ontology from natural English text. However Mohamed Yehia Daha, Hesham A. Hassan and Ahmed Rafea have still not addressed the issue of NL ambiguities, language complexity, difficulty in understanding the meaning of words and difficulty in distinguishing words which have the same meaning. Khurshid Ahmad and Lee Gillam [3] have invented an approach of ontology construction, through the automatic identification and extraction of special keywords of a domain. Raghu Anantharangachar, Srinivasan Ramani and Rajagopalan [4] have proposed an approach to populate an existing ontology with instance information present in the natural language text provided as input. Nevertheless still they have not addressed the automatic knowledge extraction from natural language text without human involvement.

SPRAT [5] is a tool developed for ontology population based on the semantic patterns. Diana Maynard, Adam Funk, and Wim Peters[5] composed the tool by combining number of GATE [6] components. Nevertheless it doesn't have strong capability to convert lengthy paragraphs other than small sentences. Hoifung Poon and Pedro Domingos [7] have presented an approach to induce and populate a probabilistic ontology using only dependency-parsed text as input. Most of them have not focused on building a system with both learning and teaching capabilities and still the machine needs more human assistance in learning.

Vanessa Lopez, Michele Pasin, and Enrico Motta [8] have presented a portable question-answering system (AquaLog), which takes queries expressed in natural language and an ontology as input and returns answers drawn from the available semantic markup. AquaLog provides the answer by extracting information from the web sites but do not explains like a human does. Then they improved it to PowerAqua [9] and PowerAqua [9] extends AquaLog [8] by sourcing answers to natural language queries by accessing any semantic document available on the Semantic Web. Nevertheless still the above mentioned limitations of AquaLog remain.

The researches on the ontology creation knowledge representation and knowledge retrieval are suffering from limitations of domain of the ontology, along with the problem of representing the actual knowledge with all related information and limitations of knowledge acquisition methodology.

The research aims to solve the problem of lack of understanding and capturing the knowledge from unstructured Natural Language sources and represent the captured knowledge through an automated approach with an effective retrieval mechanism.

II. Use of Ontology Based Environment for Semantic Learning

Knowledge extraction from NL text can be modeled using ontology driven approach. The semantic learning process deals with two major activities. Information extraction from unstructured text (IE) and represent the extracted information in ontology based environment. The research has focused on generating novel approaches over both areas while combining them to achieve the semantic learning in machine environment.

The system initiates its learning process by inputting an unstructured text document. The contents of the text documents are categorized into various levels of inputs. Inputs are categorized in to two main types named main inputs (raw inputs) and supportive inputs (additional inputs). The main inputs are unstructured plain text at paragraph level or at page level - The plain text is inserted to the system as a page which contains multiple paragraphs for the processing

- Unstructured plain text at chapter level - The plain text is inserted to the system as a chapter which contains multiple pages for the processing
- Unstructured plain text at document level- The plain text is inserted to the system as a complete document which contains multiple chapters for the processing

In addition to the above mentioned inputs the following contents of the books have used as supporting contents of the system.

- Table of contents level input – The table of content of a given book or document which includes headings and sub headings of the document
- Indexes – The indexes of a documents or a book
- References – Cited references of a given document or a book.

The research come up with a semantic learning framework which is capable of learning from the natural language sources and represent the captured knowledge using ontology based environment. The system takes various unstructured text inputs such as paragraphs, text documents, references, indexes to capture the knowledge from the NL sources. The process is automated and first it performs the natural processing with three continuous steps including Tokenization, Information extraction and semantic tagging. The system generates an ontology with the captured knowledge and which is evolvable. In this process also the human intervention is reduced and the output would be an evolvable ontology which is ready for knowledge retrieval.

The knowledge available in natural language text book will be extracted by relating the contents available in different levels of text book such as Table of contents, Chapters, and Indexes and represented in xml format with the semantics and the relationships of each content. The system consists with two sub modules which are Self Learning Module and Question based learning system-QBLS.

A. *Self-Learning Module*

The module will be designed with three main components

- Linguistic Processor
- Ontology Editor
- Updating Engine

All these modules are responsible of semantic learning and these modules will perform the basic task of capturing the knowledge from the text and represent them in an ontology based environment. The next description includes what are the responsibilities of each module in processing.

B. *Read and Process the Natural Language and Grab the Knowledge (Done by Linguistic Processor)*

Responsible of understanding the natural language and process it to create the initial knowledge of the system. Basically the processor interacts with the learning user interface of the system. Once the source document is inserted to the system, the linguistic processor validates the added source and extracts the initial information by ignoring the ambiguity and complexity of the added content. Then the extracted information will be processed by using its own experience to create the initial knowledge of the system. This is one of the main components of the system since it handles learning from the natural language text. The newly created initial knowledge is known as the ontology scratch of the system.

C. Find the Existence of Knowledge (Updating Engine)

The extracted information can be either available in the system knowledge base or it is new to the system knowledge. By depending on these two conditions the system needs to perform two activities. Either update the knowledge or create new ontology based on that information. In this scenario the updating engine is responsible of investigating the existence of the knowledge. It follows its own mechanism to determine the status of that information and it will first decide the necessary action to be done. If it is an updating content, it will analyze what parts of the existing ontology will be updated.

D. Create or Update the Knowledge (Ontology Editor)

Once updating engine finishes its task the ontology editor will receives data to create or update the relevant knowledge. Depending on the request, the knowledge will be created by adding meaningful relationships over the processed content. This task is much similar to the human knowledge acquisition process. Once it receives the relevant information, the knowledge will be created or updated. The entire process consists with defining the relationships over the different domains and also representing the knowledge in a manner which machine can process. In this operation the function includes all the activities of storing the knowledge and also defining the semantics over the created knowledge.

E. Question Based Learning System-QBLS

This is the system which performs teaching activities to its end users. The sub system is basically implemented as a question based learning system, in the sense user has to enter the question and system provides the answer with relevant explanation (as human does) along with suitable examples. In addition to that the user will be able to work on interactive learning sessions with the system.

The module will be consisting of four basic components

- Query analyzer
- Inference Engine
- Knowledge acquisition layer
- Knowledge depth analyzing engine

Each of the above components is responsible of handling the following functions in the QBLS.

F. Read the Entered Question and Process to Get the Meaning of the Question

This function is to process the question entered by the user. The function syntactically validates the user entered question and will grab the exact meaning of the entered question. Since the users are facilitated with entering questions in the form of natural language, this function will ignore the ambiguity, complexity of the question. Finally it will create the simple and exact meaning of the question.

G. Analyze the Question and Generate the Query

Once the question is converted to a simple format, this function will create the query or a format of the question which can be understood by the machine. A query language will be used for this purpose and the generated query is based on the question entered by the user and it aims to retrieve the necessary knowledge according to the user specified areas.
The generated query will be executed over the constructed ontology and retrieve the necessary knowledge from the ontology.

H. Querying the Knowledge to Find the Answer (Using the Inference)

According to the parsed query, the system should provide the correct answer .For that, the system should identify necessary knowledge components and other information required to provide the answer. Rather than just mapping the knowledge in to an answer, the system will use its previous experience to produce the answer. Explaining will take part in this process as well. Sometimes correct answer will not be provided. That means the provided knowledge is insufficient to provide the answer for the question. The function is responsible of finding the relevant knowledge components over the parsed question.

I. Knowledge Acquisition

This is responsible in getting the necessary information from the knowledge repository. Once the previous function finds the correct knowledge components, this function will retrieve the relevant information from the repositories in order to create the answer. Finally system creates an ontology capable of answering queries and evolvable with the interaction.

III. Methodology

The system contains two modules.

- Self-Learning Module
- Question Based learning module

Fig. 1 explains the top level architecture of the system along with the main modules and their functionality.

A. Implementation of Linguistic Processor

Each implementation description contains the existing capabilities of IE with the customized contents in the system. The Linguistic processor contains the following sub modules.

B. Tokeniser

Linguistic processor starts its task when the unstructured text document is inputted and that document will be sent to the tokeniser. The tokeniser splits the text into very simple tokens such as numbers, punctuation and words of different types. For implementation of this module ANNE tokeniser is used with its default tokenise rule set. The system uses the Tokeniser for different level of tokenisation.

The base level would be the sentences while a sentence can be further decompose in to different set of individual tokens.

C. Name Resolver

The role of the Name Resolver is to identify entity names in the text based on lists. The ANNIE gazetteer libraries are embedded in to the system and new gazetteer lists are created for the purpose of name recognition in the system. The gazetteer lists used are plain text files, with one entry per line. Each list represents a set of names, such as names of cities, organisations, days of the week, etc. (The default gazetteer list in ANNIE).

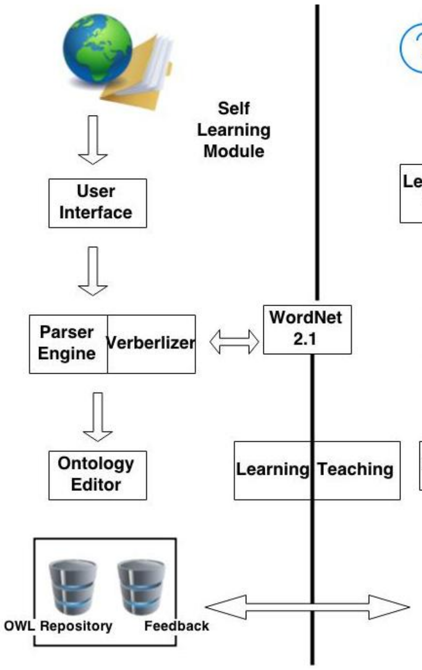

Fig. 1. Top level architecture of the system.

There are set of gazetteer lists which contain the domain specific names such as programming language names like Java, Pascal and some names related to specific algorithms in the programming language domain.In ANNIE system these gazetteer libraries which contains the lists are compiled into finite state machines. Any text tokens that are matched by these machines will be annotated with features specifying the major and minor types. Grammar rules then specify the types to be identified in particular circumstances.

D. Sentence Splitter

The sentence splitter is a cascade of finite-state transducers which segments the text into sentences. This module is required for the semantic tagger. Each sentence is annotated with the type ‘Sentence’. Each sentence break (such as a full stop) is also given a ‘Split’ annotation in the ANNIE system. The system used that feature of the IE in order to extract the sentences and separate the sentences in the given book or document. The sentence splitter is domain and application-independent.

E. POS Tagger

The ANNIE POS tagger is embedded with the system for the purpose of part of speech tagging. The POS tagger uses a default lexicon and rule set (the result of training on a large corpus taken from the Wall Street Journal). Both of these can be modified manually if necessary. Brill tagger produces a part-of-speech tag as an annotation on each word or symbol is embedded with the system in order to do the POS tagging of the uploaded document.

F. Semantic Mapper

Semantic Mapperis used to semantically tag the tokens passed by the POS tagger and based on the JAPE language. It contains rules which act on annotations assigned in earlier phases, in order to produce outputs of annotated entities. The default annotation types, features and possible values produced by ANNIE are based on the original MUC entity types plus the newly generated semantic relations through the word-net integration. According to the domain Programming languages set of JAPE Rules have been created to semantically annotate the tokenized words in the document. This annotation is the first level semantic annotation of the system based on the JAPE rules and there are three levels of semantic annotation processes of the system which describes in the next sub topics of the chapter. Sample JAPE rule written for programming languages is as follows;

```
Phase:  Programminglanguage
Input: Lookup Token
Options: control = appelt
Rule: Programminglanguage1
(
 {Lookup.majorType == programminglanguage}
 (
  {Lookup.majorType == programminglanguage}
 )?
)
:programminglanguage  :programminglanguage.Programminglanguage = {rule = "Programminglanguage1"}
```

G. Implementation of the Verbalizer

The Verbaliser is responsible of performing the second level semantic matching under three major steps.

H. Construction Process for Domain Ontology

This section applies the concept of the context assisting the construction process of Programming language domain ontology from unstructured text documents which include three processes, namely Concept Clustering, Domain Extraction, and Domain ontology scratch construction, which are described below.

This process aims to cluster concepts and instances from documents. For selecting important terms for Concept Clustering, the nouns with the highest TF×IDF values are preserved and adopted, where TF is the term frequency and IDF is inverse document frequency [10].In this process, the POS was selected as the concept similarity factors for analyzing the English terms and calculating the concept similarity between any two terms based on the features of the English language. Each node of the tagging tree denotes a POS tag defined by POS Tagger in the Linguistic processor. The path length between two nodes is adopted to calculate the concept similarity in POS between any two terms.

I. Domain extraction for Ontology Scratch

The Linguistic processor separates the text into nouns, verbs and sentences, which are then fed into the Context Extraction process to obtain the contexts. This system denotes a term as a triple (term, POS, index), where index is the position of this term in the sentence. A context is extracted if the context occurs within an interval of a given window size, and the context's occurring frequency of the text document set is larger than the defined minimal occurrence value. To increase the accuracy of the context, the punctuation is filtered and the POS of terms with Na, Nb, Nc, Nd and verbs are retained in the sentence. Context extraction algorithm is as follows.

```
/*T<t1,t2,...,tk>: The set stores the term sequence t1,
t2,...,tk
occurring in a given sentence.
T<t1,t2,...,tk>.cardinality: It denotes the number of
item in
T<t1,t2,...,tk>,and the number of occurrences of the
term
sequence t1,t2,...,tk. ti.position: denotes the position
of ti in
a sentence. sentence_num: The sequence number of
a
sentence.*/
Input: Sentences
Output: Contexts
Step 1 Generate Large 1-Sequence
For all terms ti
Scan all terms
Scan all Sentences
If ti appears in this sentence
Record sentence_num in T<ti>
If T<ti>.cardinality≥Minimal_Occurence
Mark multiple occurrences
Get Token,
Start word net
Separate in to grammar particles
Retrieve meaning t1…..tn  and add to sequence
Ignore prepositions
Calculate the semantic similarity of each word with
adjacent words
Create context similarity value
Compare with Context Rule set
Repeat with large sequence
Output context possibilities
```

J. The Domain Extraction Algorithm

The above algorithm is a modified version of ANNE IE algorithm with the word-net integration in order to identify the domain of a particular document based on the semantics exists in the content.

K. Development of the Ontology Scratch

After obtaining the context possibilities, the possible contexts are mapped to the result of the Concept Clustering to tag the concept name. Then, the Attributes, operations and associations are extracted from context according to the morphological information. Finally, the domain ontology scratches construction algorithm shows as follows:

L. Implementation of the Ontology Editor

Ontology editor uses the ontology scratch as the input and generate the ontology based on defined rule set. In this process protégé 4.1 libraries are embedded in to the system with the aid of the OWL API. Ontology web Language (OWL) is used as the ontology language and owl components are extracted from the ontology scratch which is in the XML format. Some of the basic rules used for the ontology construction process are as follows.

```
Input: Context with the concept name
Output: Constructed ontology Scratch
Step 1 For all contexts ci
Refer Gazetteer list for Name Resolving
If the number of concepts >1
If the first term t1 is an instance and the POS
Else if the Concept is NP,VP,…….Phrase
The second term t2 is sub word or adjacent of Context word
Calculate the similarity index of each context word
Compare with W-Net output
Extract the highest semantic measure.
Tag the token with semantic term
Step 2 Repeat for all sentences.
Output : Constructed ontology Scratch
```

M. OWL Rules for Ontology Construction

- If the word is a verb it is a property.
- If the word is an adjective it is a property.
- If the word is a general noun it is a class.

E.g.: Programming Languages, object oriented Languages

- If the word is a specific noun (proper noun) it can be an instance of particular class

E.g.: java, c++ (Most probably they are not dictionary words)

- If the word is a pronoun (it, he, she, they) navigate to the previous sentence' subject noun.
- If sentence contains the numeric values it can be a value a property so navigate to the same and the previous sentence and check for property identification.
- If the POS tags contains adjacent noun and a pronoun it can be a class/sub class and the instance of that particular class/sub class.

```
Set OWL Object
        Recognize OWL entities
            OWLEntity1
            OWLEntity2
 Access Class Array
   Add Entity to OWL object, OWL Class
Define the Super class of OWL Class
  Add SuperProperty
  Add OWL Property
Access Individuals Array
Create OWL Individual
Add to OWL class
Access Properties Array
Add OWL ObjectProperty
If any super properties
Add SuperProperty
Continue with Properties, Class, sub class, Instances
```

The following pseudo code illustrates the basic steps of ontology construction from the generated ontology scratch. In this process the system integrates the third level semantics over the processed content with the aid of word-net and Jena reasoning engine.

IV. Evaluation

The system was successfully designed and developed with all mentioned functions and features. The Learning module was developed using the Gate 5.1 API by overriding the existing functionality of the original component.

The newly constructed NLP builder is capable enough to identify most of the language components of the domain. The ontology construction mechanism introduced with the system was designed with 70% percent of expected accuracy and it is being tested with over 60% percent of accuracy. There were two experiments carried out with semi-automated approach and the automated approach of the system. In both approaches a document with 2 pages in the domain of programming languages was given to the 10 human experts and calculated the time taken to identify the four components of the ontology (classes, sub classes, instances and properties). The same document was processed with the system and calculated the average time taken to output the ontology with the 70% of expected accuracy. Nevertheless the system produced an output with 62% accuracy within 3 minutes and 34 seconds.

When compare to the average human expert time of 18.7 minutes the system has a significance improvement of the time taken. Meantime when we compare the accuracy of the identification the human process is more accurate than the systems approach. Further the entire ontology construction process is evaluated with various document lengths and it takes 8 to 10 minutes for document of 30 pages.

A. Statistical Analysis

The statistical analysis was based on the outcomes of the ontology construction process in terms of generating the owl ontology from the captured information from the NL text. (Generating ontology from ontology scratch)

Sample size is taken as 30 and the experiment consist of 30 different documents which have been randomly selected and the difficulty level of the document which includes the sentence length, number of tokens and the amount of information is increased.

$H_{0:}$ the overall accuracy ontology construction process (by ontology editor) is greater than or equal to 60%.

H_1: Otherwise

$$Z = \frac{\tilde{p} - p}{\sqrt{\frac{\tilde{p}(1-p)}{n}}} \sim N(0,\, 1)$$

$\tilde{p}$ =correctly identified proportion=0.55

$$= \frac{0.52 - 0.6}{\sqrt{\frac{0.52 * (1 - 0.52)}{30}}} \sim N(0,\, 1)$$

$$= 0.1857$$

Since p value > 0.05 it does not rejectH_0 and accept the hypothesis. Therefore the overall accuracy of the ontology editor in ontology construction process is greater than or equal to 60%.

V. Conclusion

The research aims to solve the problem of Lack of understanding and capturing the knowledge from Natural Language sources and represent the captured knowledge in an automated and efficient manner. The research deals with introducing a novel approach of knowledge capturing from unstructured natural language text and automated the process of representing the captured knowledge in machine understandable format. The first objective of the system was to develop an approach of learning from Natural language sources and the research has introduced a novel approach of meaningful information extraction which goes beyond the capabilities of modern information extraction systems. The knowledge capturing mechanism acquired an overall success of 65-67% based on the experiments carried out during the evaluation stage of the project.

The second objective of the research was to automate the process of representing the gathered knowledge in Ontology based platform. The system is capable of storing the captured knowledge in OWL ontology and with the question answering system it is noted that the system is capable of answering both simple and complex queries based on the captured knowledge with a 60-65% of average accuracy.

In addition to that the system introduces an expansion to the traditional Information extraction systems where IE is performed with traditional techniques plus the newly embedded semantic tagging over the extracted information.

There are several limitations and assumptions made in the project. Currently the system supports for simple question formats in English language such as WH questions, Explain, Describe. Due to the limitations of time, a necessity of narrowing down the subject domain was required. There for programming languages were selected as the subject domain. The system has suffered with the limitations of ontology querying when implementing the QBLS since lack of syntaxes in the OWL-QL has redirects the system to use the SPARQL as the query language.

As future improvements, the system is designed to work on the domain of programming languages. But it can be used as a template to expand through many other domains. Also the system can be expanded via many languages by writing the rules of the grammar of the different languages. The future work can be carried out to provide more effective services by answering to questions in verity of patterns throughout different domains. The QBLS should be updated since it is developed as an aid for evaluation purpose of the front end of the system.

The research opens a new era of mankind by introducing a teaching and learning solution to overcome the weaknesses "likeness of dependent learning" in open and distance learning. This can be promoted as a cost effective, new e-learning technology which can be used throughout the world. It provides an easy and efficient way of learning to humans without depending on a teacher or a tutor. Instead of reading an entire document, users may ask certain questions to get the required knowledge which saves time and make the learning process much easier. The system can be expanded throughout many fields such as robotics, business, medicine, etc...

REFERENCES

[1] A. M. Gadomski, "Application of system-process-goal approach for description of TRIGA RC1 system," presented at 9th European TRIGA Nuclear Reactor Users Conference, Roma: GA Technologies,TOC-19, USA, 1987.

[2] M. Y. Dahab, H. A. Hassan, and A. Rafea, "TextOntoEx: Automatic ontology construction from natural English text," presented at 06th Artificial Intelligence and machine Learning International Conference, Egypt, 2006.

[3] K. Ahmad and L. Gillam, "Automatic ontology extraction from unstructured texts," *Lecture Notes in Computer Science*, vol. 3761, 2005, pp. 1330-1346.

[4] R. Anantharangachar, S. Ramani, and S. Rajagopalan, "Guided information extraction from unstructured text," *International Journal of Web & Semantic Technology (IJWesT)*, vol. 4, no. 1, 2013.

[5] D. Maynard, A. Funk, and W. Peters, "SPRAT: A tool for automatic semantic pattern-based ontology population," presented at International Conference for Digital Libraries and the Semantic Web, 2009.

[6] *GATE-General Architecture for Text Engineering*, University of Standford.

[7] H.-F. Poon and P. Domingos, "Unsupervised ontology induction from text," in *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 296-305.

[8] V. Lopez, M. Pasin, and E. Motta, "AquaLog: Ontology based question answering system," in *Proc. Third Mexican International Conference on Artificial Intelligence*, Mexico City, Mexico, April 26-30, 2004, pp. 468-477.

[9] V. Lopez, M. Fernández, N. Stieler, and E. Motta, "PowerAqua: Supporting users in querying and exploring the semantic web content," *Semantic Web*, vol. 3, no. 3, pp. 249–265.

[10] F. Wu and D. S. Weld. "Automatically refining the wikipedia info box ontology," presented at the Seventeenth International Conference on World Wide Web, Beijing, China, 2008.

D. N. Koggalahewa was born in Sri Lanka in 1985. He received his BSc (Hons) in information technology in 2010 from Sri Lanka Institute of Information Technology, Malabe, Sri Lanaka. He completed the PGD in artificial intelligence at University of Moratuwa, Sri Lanka and currently he is reading for the MSc in artificial intelligence.

In January 2011, he joined the Department of Information Technology, Sri Lanka Institute of Information Technology as an instructor, and in June 2011, he became an assistant lecturer. His current research interests include artificial intelligence, natural language processing, ontology modelling and mobile development. He has publications in the 8th International Conference on Computer Science and Education (ICCSE 2013), 2nd research symposium University of Uva Wellassa Sri Lanka. December 2011, 30th Anniversary International Conference of Open University of Sri Lanka, August 2010 and 6th research symposium, Sri Lanka Institute of Information Technology.

Asoka S. Karunananda is a senior professor of Information Technology University of Motratuwa, Sri Lanka. At present he is the dean of Research and Development of General Sir John Kotelawala Defence University. His research interests include multi agent systems, ontological modelling, machine translation, mindfulness, and theory of computing. He has more than 100 international publications to his credit.