International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 5
ISSN: 2088-5334

Knowledge Representation Framework for Software Requirement Specification

L. Jelai[a,1], E. Mit[a,2], S. Samson Juan[a,3]

[a]*Faculty of Computer Science and Information Technology,Universiti Malaysia Sarawak (UNIMAS), 94300, Malaysia*
E-mail: [1]lilyjelai@gmail.com; [2]edwin@unimas.my; [3]sjsflora@unimas.my

***Abstract*— The need to extract correct information has become one of the main issues when analyzing the software requirement specification (SRS) documentation. The amount of gathered knowledge depends on the size of the information. However, the complexity of software systems is continuously increasing. As software systems change to more complicated systems, the information from the SRS documents may not be easily comprehended. For example, each annotation requirements tasks target the different types of information, and these tasks require the availability of experts specialized in the field. Large scale annotation tasks require multiple experts and very costly. If the number of experts is limited, annotation tasks may overwhelm the experts. The organization would not complete their objectives if they failed to manage their data because poor knowledge management affects many operations within the organization. To extract such vast information and turn it to useful knowledge, a company needs top quality software. This technology should able to input, store, and access systematically. This paper will discuss a framework based on the knowledge-based method, an attempt to improve knowledge representation. In this approach, WordNet 2.1 would be used as the knowledge source used to identify concepts represented by each word in a text from the SRS document.**

***Keywords*— knowledge-based; software requirement specification; WordNet.**

I. INTRODUCTION

Typically, the development process will include a software requirement specification (SRS) documentation. SRS documents portray a complete system behavior as it elaborates on the functional requirements, non-functional requirements, and other aspects of software systems such as business processes [1]. Thus, the achievement of a software project mostly relies on the quality of SRS documentation. SRS document helps as an input during the earlier phase, coding phase, and testing phase.

As software frameworks have been developed, software engineers ought to deal with a developing amount of data and information. According to Antunes, Gomez, and Seco [2], creating new supporting tools to support knowledge management amid software development and maintenance is essential within the software industry. This is due to the overwhelming knowledge obtain during the software development process. However, this overwhelming knowledge can be an asset for a software company. Therefore, how to make this knowledge valuable?

Antunes, Gomez, and Seco [2] recommended that every company must know how to utilize the knowledge for future reuse fully. Thus, companies should build components that can implement contexts characterization and data classification. One of the suggested ideas is to exploit the knowledge representation languages and turn it to domain conceptualizations, such as ontologies. Apart from that, these components must come out with solutions that oversee any access and exchange of important data [2].

SRS documents are frequently found to be corrupted. Most of the sentences used a full of ambiguity because it is written in an unrestricted natural language. Due to that reason, an expert must recognize and resolve any vague information manually [3]. The SRS documents can also be found in unstructured form. This situation would need additional efforts from the experts as they must extract significant information about the software. Most of these experts stated that they usually found the sentences describing functional requirements in other sections containing non-functional requirements and vice versa. To understand the differences between non-functional requirements and functional requirements, according to Hussain, Ormandjieva, and Kosseim [4], the non-functional requirement is a software requirement that articulates the quality requirements and the constraints over the related behavior of the system. For example:

> *"All the mandatory attributes cannot be empty, and the budget amounts cannot be negative"* [4].

Meanwhile, a functional requirement is defined as a software requirement that articulates the required behavior

of the system [4]. For example:

"*User finally saves the budget*" [4].

However, with the enormous development of software nowadays, do one company needs to hire more expertise to manage various information? According to Sateli at. Al [5], the vital activities in software development is the annotation of software requirements. Thus, as stated by Sateli at. al [5], it is an unavoidable situation and these experts must deal with nonchalantly written SRS documents during the requirements specification phase [5], which include extracting vague information and noise in sentences that do not represent any types of software requirement [4].

Even though SRS documents can be reused, the annotation requirements process should follow a standard taxonomy requirement [6]. This is because every software has different information and annotation requirements. Hence, a company must have many experts on a different type of requirements annotation tasks. These experts also should always available for all projects. Hiring extra expertise would increase overall project costs.

With the advancement in current technology, knowledge management of software requirements has become one of the essential studies among researchers. Generic knowledge with specialty fields and the project application domain were included in the knowledge management [7]. Most importantly, this knowledge mechanism must be able to capture information from the former and alike projects [7].

This paper is to propose a framework that uses knowledge-based to form a knowledge presentation based on SRS documents according to their respective system. The knowledge-based method will use Wordnet and domain databases containing information from various domains. To identify the sense of words in a context [8], this proposed framework will use knowledge resources like Wordnet so that it can produce a better knowledge representation because a knowledge-based approach exploits a vast amount of structured knowledge. The motivation for suggesting this framework is to create a conceptual knowledge representation from the SRS documents. The advantage of using conceptual knowledge representation is to assist in identifying the possible outcomes of SRS documents according to the respective domain.

A. Wordnet

Wordnet has been used widely as preferable lexical resources [9], [10]. Unlike other dictionaries, the information systematized in WordNet is clustered into sets of cognitive synonyms. Meanwhile, each synonym expressing a distinct concept. Features such as part-of-link and the antonym links were included in the database [11].

TABLE I
CURRENT DATABASE STATISTICS IN WORDNET 2.1

Part-of-Speech	Unique Strings	Synsets	Total word-sense pairs
Noun	117097	81426	145104
Verb	11488	13650	24890
Adjective	22141	18877	31302
Adverb	4601	3644	5720
Total	155327	117597	207016

The current database statistics in Wordnet 2.1 are shown in Table 1. Since Wordnet consists of super-subordinate relation [12], Wordnet is implemented as knowledge-based in this framework because almost all general synsets as features in WordNet are linked. According to Fellbaum [12, and all noun hierarchies in the Wordnet database are ultimately gone up the root node.

Most of the WordNet's relations connect words from the same part of speech (POS). Hence, WordNet consists of four sub-nets, one each for nouns, verbs, adjectives, and adverbs, with few cross-POS pointers [10], [12]. Cross-POS relations include the morphosemantic links that hold among semantically similar words sharing a stem with the same meaning. For example, the word *observes* (verb), *observant* (adjective) *observation, observatory* (nouns) are used by Fellbaum [12] to set an example that shows links between similar words semantically. This kind of feature would enhance the creation of knowledge later discussed in Section II.

B. Related Works

Chikh [7] developed a combination of conceptual framework and knowledge creation by using the "Socializer"; "Externalizer"; "Combiner"; and "Internalizer" (SECI) model and domain ontologies to oversee the Software Requirements Engineering (SRE) adapting the knowledge-based approach. These subsystems are "Socializer;" "Externalizer;" "Combiner;" and "Internalizer" (SECI) implemented innovative features by allowing key actors to involve in software requirements processes like elicitation, specification, and validation in one frame interactively. All subsystem "Socializer"; "Externalizer"; "Combiner"; and "Internalizer" (SECI), according to Chikh [7] have their corresponding repository. The function of each repository is to create interactive knowledge. Apart from that, the attached repository is expected to create or exploit knowledge assets or do both simultaneously. To adapt semantic research, these subsystems are also connected to domain ontologies, which are Application Domain Ontology (ADO) and Software Requirements Ontology (SRO).

Antunes, Gomes, and Seco, introduces a Semantic Reuse System (SRS) [2], a system that allows users to reuse software development knowledge. This system exploits semantic web technology such as Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), and Web Ontology Language (OWL) to represent the knowledge used by the system. Various software development process elements like specification documents and design diagrams were considered in this study—each one of these elements named as a Software Development Knowledge Element (SDKE). According to Antunes, Gomes, and Seco [2], the main objective of this system is to provide proficient components by integrating the Semantic Web languages that able to store, search, retrieve and manage the knowledge.

To extract semantic relation from text documents, Ta and Thi [13] integrated both statistical methods and natural language processing. Ta and Thi [13] only used ACM Digital Library text documents, and these documents would classify automatically when applying stated methods. Ta and Thi [13] stated that this approach is separated into two core

modules. The first module is Computing Domain Ontology (CDO), which undergoes the statistical method. Meanwhile, Wordnet and other dictionary resources were used to classify the semantic relations among the instances in another module. According to Ta and Thi [13], the documents used in this study are based on the computing domain [13]. Both modules will produce ontology that shows instances with their respective semantic relations from text documents. Nevertheless, as stated by Ta and Thi [13], not all information can be extracted by the modules.

To undergo the annotation process of text documents in a natural language, Hassan et al. [14] applied semantic technology. Therefore, this study is concentrating on finding the meaning of sentences. This study also focuses on identifying the possible requirements in the text documents. [14]. Hassan et al. [14] propose four modules, which are Data Cleaning, Graph Construction, Sparse Matrix, and Ontology Construction. The drawback of this study is that it does not convey any knowledge presentation.

Gaeta et al. [15] focused on the creation of knowledge in the form of ontology from the heterogeneous text. This study explained the combination of five modules, which are Pre-processing, First Ontology Creation, Concept and Relationship Creation, Harmonization Refinement, and Validation. The heterogeneous text will use in the modules to generate ontology. The generated ontology consists of concepts and relationships between words. As a result, the generated ontology can deliver a contextual understanding of a certain text. However, Gaeta et al. only applied this system to heterogeneous text.

Jelai et al. [16] proposed a framework to create a presentation of knowledge from the text requirements. This study is using a knowledge-based word sense disambiguation approach. A combination of knowledge-based word sense disambiguation approach and WordNet 2.1 were used in this research to create knowledge from text requirements during the analysis phase. The proposed framework consists of four modules, which are Term Extractor module, Pre-Processing module, Knowledge Builder module, and Knowledge Representation module. However, the presentation of knowledge in this study is merely based on any text requirements without a specific domain because the proposed framework is not domain-dependent.

II. Materials and Method

The proposed framework in this paper is elaborated in this section. The framework will use WordNet 2.1 and the domain database as the knowledge source.

The module to create knowledge in this study is based on Ta and Thi [13], which is obtaining knowledge from documents [16], and this work extends the requirement module from normal text document to software text requirement. Figure 1 below shows a framework that has several modules. Each module will use results from prior modules.

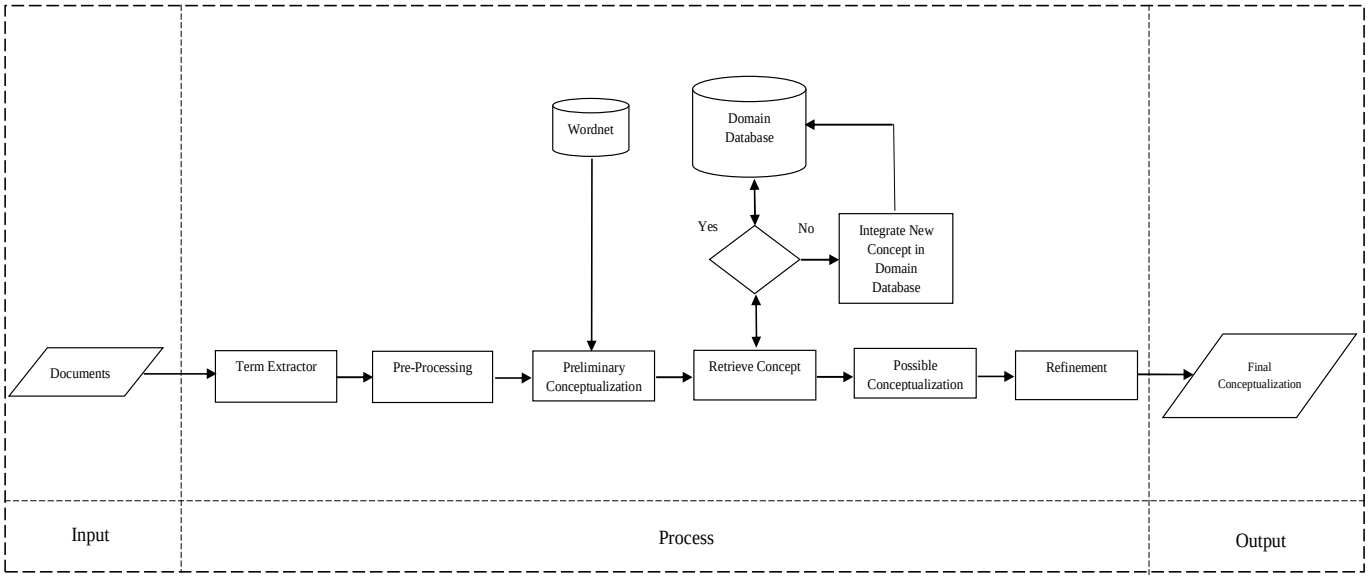

Fig. 1 Knowledge-based method for Knowledge Representation of SRS

The intended outcomes of this research are stated as follows:

- Knowledge representation of SRS document and
- The knowledge produced will have a conceptual relation between terms.

The followings are the descriptions of every module.

A. Input

Any SRS document files with a various formats like DOC, PDF, and TXT will be considered as input of the framework.

B. Process

Several sub modules would be described in the following:

1) Term Extractor: This sub-module is implementing the method used in [15]. The purpose of this sub-module is to select the relevant term that could be found from the SRS document. The selection task will comprise a set-of-term filter, which allowed the submodule to calculate a score for a term relevance used in SRS document. Only the terms with a score over a given threshold are considered as relevant.

According to Gaeta et al. [15], the calculation in this sub-module is significant to dodge missing document information.

2) Pre-Processing: This sub-module will reduce terms that are found from the results obtained from the Term Extractor. To reduce the term to its root, a combination of algorithms is implemented [16]. The root is important so that the word appears in an appropriate form. For example, the word "reading" is reduced to "read." In this submodule, we propose stemming, part-of-speech tagging (POST), and stopword list method. Stemming is the method that will reduce the term either to its stem or root with a combination of the algorithm. Part-of-speech tagging (POST) is a submodule classifies the terms obtained from the SRS document to a part of speech. Terms could consist of names and verbs. The Stopword list comprises removing the irrelevant terms inside SRS document.

3) Preliminary Conceptualization and Wordnet: The knowledge created will using HasPart (HP) relation as it is adapting the ontology characteristics, assuming the connection between words inside the SRS documents are semantically annotated. To see the words are semantically annotated to one another, HP's relation is the best choice. HP relation can provide a complete relationship among the words from SRS documents. A synonymy graph concept [17] will be implemented in this module to create the first relation. Since Wordnet 2.1 can identify the concept represented by each term inside the SRS documents, so Wordnet 2.1 will be used once the knowledge representation is ready. Useful semantic correlation among the concepts also one of the features provided by Wordnet [15], [18], and this is fully utilized in this submodule.

4) Retrieve Concept and Domain Database: This sub-module presents the reasoning mechanisms used for manipulating the preliminary conceptualization and knowledge that are stored in the domain database. Operations such as suggest, search, and retrieve knowledge from the database would be implemented right after the concept from the previous section has been retrieved. If the concept exists, any related knowledge from the database would be added to the preliminary conceptualization.

5) Possible Conceptualization: This sub-module presents the preliminary conceptualization obtained by the previous submodule.

6) Refinement: This is the final submodule in the process component. The purpose of this sub-module is to refine output from the previous module by detecting anomaly on the retrieve concepts. An example of anomaly could be the overgrowth deepness of some concepts in the graph concerning the average depth [15]. The user must solve any detected anomalies. One of the suggestions is by allowing user to modify the conceptual structure by modifying the text requirement that is associated with the concept.

7) Integrate New Concept in Domain database: Submission and categorization of new knowledge will be done in the Integrate New Concept module. Any new knowledge will be indexed to concepts and store in the database for future use. Integrating new concepts only be used if the preliminary concept from the previous module does not exist inside the database. A couple of processes should be done to index the new concept into a domain database. The first process is to extract the information from the SRS documents. After that, the system must verify the existence of the concepts in the domain database. Users should consider imported the concepts from additional sources if the concepts do not exist.

C. Output

Final Conceptualization: This final module is to show the entire knowledge representation obtained from SRS documents. A graph will present the extracted knowledge. This graph contains several relevant nodes, as shown in Figure 2. Based on the graph in Figure 2, the nodes C, C1, and C2 are the possible conceptualization: signified by the node T meanwhile nodes S as in the present the senses obtained from WordNet 2.1.

The knowledge representation, as shown in Figure 2, only displays a graph of one relevant term with their respective concepts and senses from the SRS document. Once the final conceptualization of SRS document is completed, a wider knowledge representation will be shown. This paper is to contribute knowledge representation based on SRS documents according to their respective system using Wordnet and domain database that store information from various domains. The main purpose of creating this framework is to build a conceptual knowledge representation from the SRS documents that able to predict several important requirements based on the knowledge that has been created.

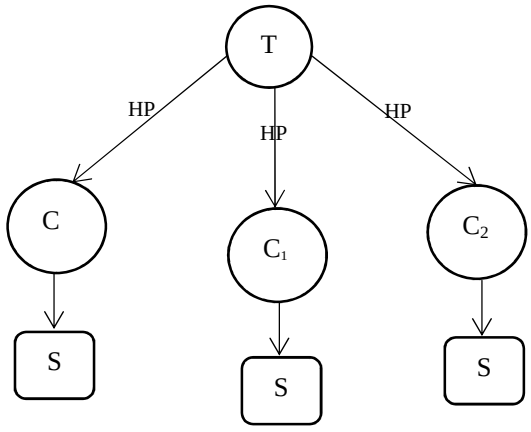

Fig. 2 Example of HP relation for a term

III. Results and Discussion

To evaluate the proposed framework, this paper will use an ontology evaluation approach by Brank, Grobelnik, and Mladenić [19]. According to [19], there are several approaches available. These approaches are described as follows.

A. Golden Standard

In general, the gold-standard approach does provide a method to evaluate ontologies. However, this approach has its own limitation. The first step when using this approach is to evaluate the golden standard itself, as stated by Brank, Grobelnik, and Mladenić [19]. Therefore, it is difficult to

determine the quality of the golden standard and to identify the source of the errors. Usually, the difficulties involve incorrect golden standard or corrupted results.

B. Application-based

The application-based approach is to measure the effectiveness of an ontology in the context of an application. According to Brank, Grobelnik, and Mladenić [19], one of the disadvantages using this approach is, this approach can be applied in one application context; the others may not. Therefore, generalizing the task-based evaluation will become a major limitation when using this approach. According to Brank, Grobelnik, and Mladenić [19], if this approach is used in an automated setting with a variable number of ontologies, the results would become unmanageable. However, any small set of ontologies is recommended to use this approach [19].

C. Data-driven

This approach involves a comparison between the existing data in the domain and the ontology(*ies*). However, this approach is not preferable when domain knowledge is concerned because domain knowledge becomes constant major limitation if using this approach [19].

D. Human Assessment

According to Brank, Grobelnik, and Mladenić [19], this approach involves ontology evaluation through users' experiences. One of the disadvantages that could be happened when using this approach is difficult to establish objective standards where the criteria (metrics) for evaluation is concerned—other disadvantages when involving human assessment such as the establishment of the right users.

TABLE II
A SUMMARY OF APPROACHES TO ONTOLOGY. SOURCE: [19], [20]

	Approach to evaluation			
Level	Golden Standard	Application-based	Data-driven	Assessment by human
Lexical, vocabulary, concept, data	x	x	x	x
Hierarchy, taxonomy	x	x	x	x
Other semantic relations	x	x	x	x
Context, application		x		x
Syntactic	x[1]		x	
Structure, architecture, design				x

Table 2 shows levels and approaches, according to Brank, Grobelnik, and Mladenić [19]. 'X' indicates the availability of the approaches used at these levels [19]. Meanwhile, X^1 indicates a "Golden standard" between the syntax in the ontology definition and the formal language syntax specification (e.g., RDF, OWL) [19] when the comparison is made. Every function of each level is explained as the following, according to Hlomany and Stacey [20].

E. Lexical, Vocabulary, or Data Layer

According to Hlomany and Stacey [20], evaluation at this level involves comparisons with various sources of data with regards to the problem domain and techniques such as string similarity measures. This level also focuses on different types of concepts that have been included in the ontology. Apart from that, this level also focuses on the vocabulary used to represent or identify those concepts. [20].

F. Hierarchy or Taxonomy

Hierarchy or taxonomy is a relation hierarchy is the most general concept in the ontology. This *is-a* relationship is a very significant concept because it can portray specific evaluation measures according to Hlomany and Stacey [20] even though various other relations between concepts are defined.

G. Other Semantic Relations

According to Hlomany and Stacey [20], these relations is evaluated separately (separated from *is-a* relation). The preferable measures in this level are precision and recall [20].

H. Context or Application Level

Hlomany and Stacey [20] stated that one ontology could be part of other large collections of ontologies. This ontology also can be referenced by various definitions in other ontologies and vice versa. So, this context is critical during the evaluation. Another level that should consider is the application where the ontology is used. The focus should be more on ontology usage and how it would affect the application results [20].

I. Syntactic Level

According to Hlomany and Stacey [20], evaluation at this level is for manually constructed ontologies by users. Formal language is used in this syntactic level. Hlomany and Stacey [20] also stated that all syntactic requirements must complement the formal language. Besides that, the presence of natural language documentation also should take in considerations.

J. Structure, Architecture, Design

According to Hlomany and Stacey [20], this level is to ensure all pre-defined principles or specific criteria are fulfilled by the ontology. One of the principles are structural concerns about the ontology and its compatibility for further enhancement. This level usually proceeds entirely manual [20].

Based on our observation in Table 2, we will use the "golden standard" approach. Precision and recall concepts [21], would be used to evaluate the lexical content in SRS documents. *Precision* is the percentage of the lexical entries (strings used as concept identifiers in the SRS documents) that also appear in the golden standard, relative to the total number of words. Meanwhile, *Recall* is the percentage of the golden standard lexical entries, relative to the total number of golden standard lexical entries. The calculations are shown as follows:

$$\text{Precision} = \frac{A}{A+C} * 100\% \quad (1)$$

$$\text{Recall} = \frac{A}{A+B} * 100\% \quad (2)$$

where:
A = Relevant words counted in SRS document
B = Relevant words not counted in SRS document
C = Irrelevant words counted in SRS document

IV. Conclusion

This paper discussed our proposed framework for knowledge representation of software requirement specification. The proposed framework combined the use of the knowledge-based method, WordNet, and domain knowledge to produce a knowledge representation of the SRS document. Developers could use the knowledge representation to predict several crucial software requirements according to a specific domain. For future work, the proposed framework will be developed into a working tool. It will be tested with the software requirement specifications from various domains. To evaluate the knowledge representation, a "golden standard" approach will be used.

Acknowledgment

The authors would like to acknowledge Universiti Malaysia Sarawak for providing facilities to conduct this research and reviewers for their comments to improve this paper.

References

[1] M. Kamalrudin, and S. Sidek, "A review on software requirements validation and consistency management," *International Journal of Software Engineering and Its Application*, vol. 9, no. 10, 2015.

[2] B. Antunes, P. Gomes, and N. Seco, "SRS: a software reuse system based on the semantic web," *In 3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, June 2009.

[3] I. Hussain, O. Ormandjieva, and L. Kosseim, "Automatic quality assessment of SRS text by means of a decision-tree-based text classifier", *in Quality Software Seventh International Conference*, 2007, pp. 209-218.

[4] I. Hussain, O. Ormandjieva, and L. Kosseim, "Lasr: A tool for large scale annotation of software requirements," *in Empirical Requirements Engineering (EmpiRE), IEEE Second International Workshop*, Sept. 2012, pp. 57-60.

[5] B. Sateli, E. Angius, S.S Rajivelu, and R. Witte, "Can text mining assistants help to improve requirements specifications", *Mining Unstructured Data (MUD)*, 2012.

[6] A. Rashwan, O. Ormandjieva, and R. Witte, "Ontology-based classification of non-functional requirements in software specifications: a new corpus and svm-based classifier", *in Computer Software and Applications Conference (COMPSAC), 37th Annual*, July 2013, pp. 381-386.

[7] A. Chikh, "A knowledge management framework in software requirements engineering based on the SECI model", *Journal of Software Engineering and Applications*, vol. 4, no.12, pp.718, 2011.

[8] N. Roberto, "A quick tour of word sense disambiguation, induction and related approaches," *in Proc.SOFSEM 2012: Theory and Practice of Computer Science Conf*, 2012, pp. 115-129.

[9] C. F. Baker, and C. Fellbaum, "WordNet and FrameNet as Complementary Resources for Annotation," *in Proc. 3rd Linguistic Annotation Workshop Conf.*, 2009, pp. 125-129.

[10] F. Fabbrini, M. Fusani, S. Gnesi and G. Lami, "The Linguistic Approach to the Natural Language Requirements Quality: Benefit of the Use of an Automatic Tool," *in Proc. 26th Annual NASA Goddard Software Engineering Workshop*, 2001, pp. 97-105.

[11] K. Knight and S. K. Luk, "Building a Large-Scale Knowledge Base for Machine Translation". *AAAI*, vol. 94, pp. 773-778, Oct.1994.

[12] C. Fellbaum, WordNet: The Encyclopedia of Applied Linguistics. *John Wiley & Sons, Ltd*, Nov. 2012.

[13] C. D. Ta and T. P. Thi, "Automatic Extraction of Semantic Relations from Text Documents," *in International Conf. Future Data and Security Engineering*, Nov 2016, pp. 344-351.

[14] T. Hassan, S. Hassan, M.A. Yar and W. Younas, "Semantic analysis of natural language software requirement," *in 6th International Conf. Innovative Computing Technology (INTECH)*, Aug. 2016, pp. 459-463.

[15] M. Gaeta, F. Orciuoli, S. Paolozzi, and S. Salerno, "Ontology extraction for knowledge reuse: The e-learning perspective," *IEEE Trans. Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol.44, no.4, pp. 798-809, Jul. 2011.

[16] L. Jelai, E. Mit, S. F. Samson Juan, and W. S. Cheah, "Textual Analysis by using Knowledge-based Word Sense Disambiguation Approach", Vol. 9, Iss. 3-3, pp. 159-162, 2017.

[17] Y. Shin, Y. Ahn, H. Kim and S.G. Lee, "Exploiting synonymy to measure semantic similarity of sentences," *in Proc. 9th International Conference on Ubiquitous Information Management and Communication*, Jan 2015, pp. 40.

[18] C. Fellbaum, "Wordnet(s)" *in Encyclopedia of Language & Linguistics, 2nd ed*. vol. 13, Keith Brown. Oxford: Elsevier, 2006, pp. 665-670.

[19] J. Brank, M. Grobelnik, and D Mladenić, "A survey of ontology evaluation techniques", Conference *on Data Mining and Data Warehouses, Ljubljana, Slovenia*, 2005.

[20] H. Hlomani, and D. Stacey, "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey.", *Semantic Web Journal*, vol. 1, no. 5, 2014.

[21] J. Ma, W. Xu, Y.H. Sun, E. Turban, S. Wang and O. Liu, "An ontology-based text-mining method to cluster proposals for research project selection," *IEEE Trans. Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 3, pp. 784-90, May 2012.