Automated Knowledge Understanding And Recognition Assistant (AKURA)

H.H.N.C.Jayanandana[1], R.Rishanthakumar[2], H.P.N.H.Herath[3], H.M.S.Piyasundara[4], D.N.Koggalahewa[5]
SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY, SRI LANKA
nilesh.jayanandana@yahoo.com[1], rishanthakumar@gmail.com[2], nipunaherath@hotmail.com[3], madumal.piyasundara@gmail.com[4], darshika.k@sliit.lk[5]

***Abstract* – Unstructured data gathered from Internet sources can be used to derive useful information from analyzing the data. The challenge is making these unstructured data to be understood by machine programs and to get the most use out of it. Nowadays techniques such as text mining, natural language processing, and natural language understanding are used to process the unstructured data. Knowledge extraction from the unstructured text should follow a standard flow of steps in order to make machines understand the data. This research involves addressing the difficulty in automating the integration of knowledge models extracted by Natural Language Processing and Understanding through software in a dynamic machine environment. This document proposes and describes of a platform, which would enable automating knowledge understanding and integration into an ontology model, which could then be used to develop a domain specific product by simply giving unstructured textual data as the raw input. The approach to this project involves with a simulation of Natural Language Processing and Understanding, which is similar to humans, by using an ontological approach and representing the extracted data into a knowledge base, which could later be retrieved in an automated way. This will be done by semantic analysis and identification of relationships between tokenized streams of information. The generic platform proposed in this document will allow users to develop a constantly evolving knowledge model by processing unstructured human-readable forms of data without any human interaction. This platform will allow organizations to develop a well organized, meaningful and a well-structured domain specific product according to their requirements. Several techniques and methodology are used such as Natural Language Processing and Understanding, parsing JSON to OWL, ontology merging and knowledge retrieval using JENA and SPARQL.**

***Keywords* – Natural Language Understanding, Web Ontology Language (OWL)**

I. INTRODUCTION

Unstructured data is information that either does not have a pre-defined data model or is not organized in a predefined manner. Unstructured data is typically text-heavy and requires human intervention in order to extract meaningful information [1]. A simple use case to describe the above issue, is the need of a product owner to get a statistical analysis of feedbacks and reviews provided by consumers to identify the issues regarding a particular product. If we are to process these reviews manually, it will be time consuming and expensive. In a situation like this, it would be highly efficient if the process could be automated without human intervention. Technologies like Natural Language Processing (NLP), Ontology concepts and self-learning concepts are used heavily to identify and extract the semantics in unstructured data.

In processing unstructured data to get useful information, most of the organizations and institutes face difficulties in learning and applying the above mentioned technologies to their business domain. As a result, many organizations opt to leave the idea and pursue alternative, less effective methods. In order to overcome this problem, we have come up with a research on developing a platform where it can automate the processing of unstructured data to useful information by providing a set of tools which could help the developers to reduce the development time and achieve their business goals. There are three main tools, which will be provided through this platform as mentioned below.

- NLU Engine
- JSON TO OWL Mapper
- Ontology Merger

Natural Language Understanding (NLU) engine is a tool, which is capable of generating tagged information and semantic details into a hierarchical data structure when raw text is provided as an input. This tool identifies the main entities in a given text paragraph, overall domain according to the context of the paragraph. Importance of the entities or the features identified according to the paragraph involved. JSON to OWL Mapper is a tool provided by our platform, which is used to generate ontologies in OWL format from any given hierarchical data structure. This tool is created to overcome the problem where data represented in a JSON [29] format being unable to represent semantic information. Output generated from this tool can become the input to our next tool, the Ontology merger. Semantic information stored in a knowledge base should be able to evolve and expand when new information is provided. In order to achieve this, multiple knowledge bases should be able to merge and the existing knowledge base should evolve. Ontology merger is capable of merging two ontologies and evolving the existing knowledge base. All these tools can be used separately without depending on each other, where it enables the users to use our tools according to their requirement.

II. RELATED WORK

The platform proposed in this document doesn't exist in any form of research yet with all three components explained below integrated together as one. Therefore, the summary of the information we extracted from multiple research papers will be included in the below sections of this document.

A. Natural Language Understanding Engine

We found quite a few researches in the field of Natural Language Processing and Understanding. In [4], a journal published in International Journal of Multimedia and Ubiquitous Engineering, the research suggests using a "Domain Dictionary" as the best technique for Information Extraction because it accesses the core part of the word pattern and analyzes the theoretical properties of the word. In this research, they consider three ways of Information Extraction in order to compare and contrast each other. The three techniques they compare are Stemming, Domain Dictionary and Execution List. Here, the group categorizes stemming into two sub techniques as Derivational and Inflectional stemming where Derivational Stemming derives a new word by simply changing it's grammar. They describes Inflectional stemming by quoting "When the normalization is confined to regularizing grammatical variants such as singular/plural or past/present, it is referred to inflectional stemming" [5] from a journal published by Manu Konchady.

E.g.: verify-verified-verifies, walk-walked-walks

According to above example, in both the cases, all the words in the example will be treated as 'verify' and in the second example, will be treated as 'walk'. The next technique they considered is "Domain Dictionary" method. This method simply uses a Knowledge Base, which consists of a collection of 'feature terms'. The dictionary structure is further divided into three categories namely, Parent Category, Sub-category and Word. The 'Parent Category' is a set of words, which are unique, and 'Sub-category' may inherit multiple parent categories and then the words inherit the 'Sub-Category'. The last of the techniques they considered is "Exclusion List", where an exclusion list or an unwanted words list is maintained separately containing words such as the, a, an, if, off, on etc. and extracting only the 'wanted words'.

B. JSON to OWL Mapper

In [6], they propose an OWL extraction method for JSON data. It explores latent schema information behind JSON objects based on semantic web technology, and further annotates and reformats JSON data to OWL ontology.

C. Ontology Merger

In [15], Semi-Automatically Mapping Structured Sources into the Semantic Web, a journal published in 2012 by the developers of Karma Framework explores expanding an ontology using existing databases or data stores with a semi automatic approach. As the first step, they assign semantic types, which involves mapping each column of the source to a node in ontology. This is not automated and is a user-guided process where the user guides the system on assigning the types using a GUI provided. In the next step a graph is constructed that defines space for all the mappings between source and ontology. Third step refines the graph based on user input. The graph is constructed so that the mapping between source and ontology can be computed using Steiner tree algorithm [16].

The researchers Enrico and Antonio in [17] have approached ontology integration for ontology reuse with implementing the following components in their framework. In order to search for proper data models, it is needed to identify the knowledge domain and the related subdomains covering the specific topic under study. Next component is the Reference Model Reconciliation and Normalization function block. It is responsible for obtaining an alignment which a set of correspondences between the matched entities from the reference models. It involves three types of matching operations: string, linguistic and extended linguistic matching. Last component of the framework is the Reference Models Merging or Integration function block.

Fisnik Dalipi, Florim Idrizi, Eip Rufati, Florin Asani in their research, On Integration of Ontologies into E-learning Systems [18] tries to implement a better solution for organizing and visualizing didactic knowledge. Their paper aims at proposing a model, which is focused on integrating ontological principles with e-learning standards. They have developed a prototype model that is integrated with ontology, which gives a semantic representation of learning contents by adding semantic notations to each learning resource. They have used the Resource Description Framework (RDF) as the underlying technology in achieving their target on the specified domain of e-learning.

III. METHODOLOGY

A. Natural Language Understanding Engine

In the proposed NLU engine, unstructured textual data is provided as an input and the following processing stages will be executed to produce the expected output.

a) Data Segmentation
b) Pre-Processing
c) Entity Extractor
d) Noun Sequence Entity Extractor
e) Domain/Category Identifier
f) Pronoun Replacer
g) Relationship and Feature Extractor

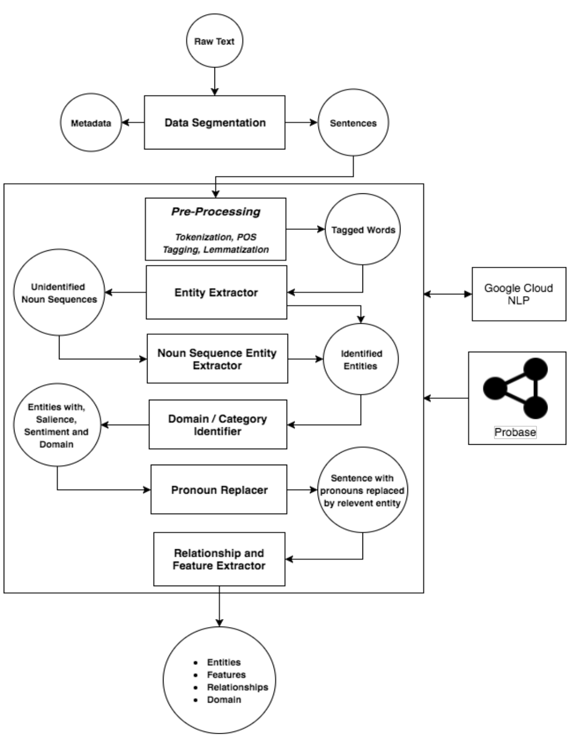

Figure 1: High Level Diagram of NLU Engine

a) Segment Relevant Data

In this phase, the input will be a data object which carries several metadata and the main sentences or paragraphs to be processed. The metadata is separated from the input data chunk and kept aside while sentences / paragraphs are passed to pre-processing stage.

b) Pre-Processing

This is the stage where sentences or paragraphs will undergo basic NLP processing such as tokenization, POS tagging, lemmatization, etc. Google NLP API [30] is used to carry out the above NLP processes. As the input to the next phase, this gives a collection of sentence wise tagged words.

c) Entity Extractor

In this phase, possible entities are identified by using the Google NLP API where it gives a collection of entities with a salience score and sentiment score. Salience is a score, which gives the importance of the entity according to the sentence or to the whole paragraph based on the input. In this process, some of the possible noun sequences which are not identified as entities by the API will be extracted separately in order to process it further to identify whether it contains entities which can be identified by a human, but not by the API. Therefore in this stage the main outputs are the identified entity list and a collection of noun sequences. Where the collection of noun sequences will undergo the Noun Sequence Entity Extractor phase.

d) Noun Sequence Entity Extractor

From the identified noun sequence collection of the previous stage, this phase will identify the possible entities out of them. For this, the identified noun sequence collection will process through the ProBase [32] ontology database and find out the meaningful domains of each and every identified noun sequences. After identifying the domains of these noun sequences it merges with the previously identified entity collection from Google NLP API. Therefore the output of this phase is a collection of fully identified entities with their scores and domains.

e) Domain / Category Identifier

In this phase, it is identifying the domain or the category which is described throughout the sentences / paragraph by Google NLP API. It gives several possible domains / categories which these sentences can fall into and a confidence score which can be described as the accuracy of prediction of such a domain / category. Therefore, from this phase it will provide the domain / category which has the highest confidence / accuracy level as the output.

f) Pronoun Replacer

The process of this phase is replacing the possible pronouns of the sentences / paragraph without making a change to the meaning of it. In this process, following assumption is made in order to find the entity and replace the pronoun. The entity which is referring from a pronoun is the entity which has the highest salience in the previous sentence or before the pronoun occurs. Therefore, as a preprocess of this component, the paragraph will be tokenized by sentences and find out the main entity of each sentences. After pre processing happens it will replace the starting 'It' of the immediate sentence by the identified main entity of the previous sentence. As the output of this phase, it is generating the new sentence / paragraph with the replaced entities.

g) Relationship and Feature Extraction

This phase of the NLU engine is identifying the features and relationships among the features and the entities identified by previous phases. Let S = $\{s_1, s_2, s_3, \ldots s_n\}$ be a set of sentences, each sentence s in S is a set of words from a vocabulary W = $\{w_1, w_2, w_3 \ldots w_n\}$, i.e: $s \subseteq W$. Let FR denote a set of main features found in the sentence collection after applying pattern mining on set of sentences S. FR = $\{f_1, f_2, f_3, \ldots. f_n\}$ is the list of main features extracted. In order to measure the degree of relatedness of each word from the set of words(w) to the main features (f) FR $\in$ W, the distance between the word (w) and feature (f) in sentences are used. Let R(w,f) be the set of relationships in the sentence collection (S) contains w and f; R(w,f) = $\{r(w,f)_1, r(w,f)_2, r(w,f)_{3, \ldots} r(w,f)_n\}$. Therefore the output of this phase is a set features and relationships among the features and corresponding entities. This output will be the major outcome of this NLU engine and it is passing a JSON object with the identified entities, features, relationships and sentiment scores to the next component.

B. JSON To OWL Mapper

A hierarchical data structure in JSON format is provided as an input to this component. There are three main data types identified in a given data structure as literal type (lt), object type (ot) and array type (at). The entire JSON object given as an input is treated as an object type (ot). Let K = $\{k_1, k_2, \ldots, k_m\}$ be a set of keys in an *object type (ot),* each Key k points to one of the types T = {*lt, ot, at*} where *lt* has a literal value, *ot* has key value pairs and *at* is an array of either *lt, ot* or a combination of both. The algorithm will recursively go through each of the keys in *ot* and *at* till every key end in an *lt.* In each recursive iteration of keys in an *ot,* Entities E = $\{e_1, e_2, \ldots, em\}$ are derived per recursive iteration and the hierarchical relationship is preserved between e_i and $e_{(i+1)}$ by taking the key as the relationship which plays a role in linking the two Entities semantically. When all the Entities are created according to the data structure provided by the input, for each Entity E, is checked with the class structure of an Ontology, which will be created at the beginning of the process. The class structure C = {} in the first iteration of looping the Entities E, and will gradually increase in size when e_i is not matched with an element in the set C. Once the Entity iteration is over and the class structures are defined in the set C = $\{c_1, c_2, \ldots c_m\}$, c_i is then analyzed with rest of the elements in the set to find structural matches and should a structural match is common between two elements in set C, a superclass is created in the Ontology making the elements in question child classes of it. Once this is done, the ontology structure is complete and all the semantic information derived in each Entity in set E will be available in the ontology.

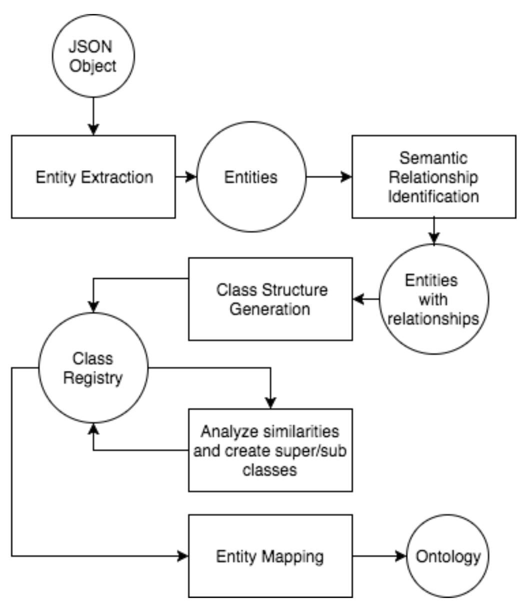

Figure 2: High Level Diagram of JSON To OWL Parser

C. Ontology Merger

Ontology integration is a complicated process done either by hand, semi automatic tools or full automatic tools. The methodology proposed in this section aims to automate the ontology integration and mapping process in a generic way, which will be applicable to multiple domain segments in a dynamic machine environment. When 2 ontologies O_1 and O_2 are given as an input, a Class Registry is created which is a set of classes $C = \{c_1, c_2, \dots, c_m\}$, where c is an object representing the structural details of a class belonging to either O_1 or O_2. Then, for each element in set C, the structural similarities are observed with the rest of the elements in the same set and super-classes and sub-classes are created accordingly between the class objects that has similarities.

Once the classes are defined in the ontology, the next step is to merge the semantic information exist in O_1 and O_2 as class instances or individuals. In this step, semantic information duplication is prevented and semantic information evolution is enabled by allowing the differences in a certain instance to merge into Ontologies O_1 and O_2. In order to achieve this, the machine should understand whether an instance in each O_1 and O_2 are semantically the same. This is being identified by using a hash value generated using the string hashing algorithm provided in java class *java.lang.String* with the most common similar word taken from thesaurus Words API [2] related to calculate the semantic similarities in two instances or individuals by taking the values of each instance in question. As the final step, after all the verifications, the semantic information contained in O_1 and O_2 will be merged and a new Ontology O_m is provided as the output.

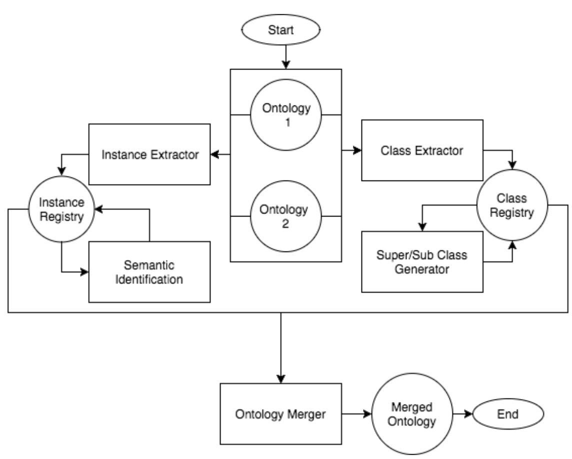

Figure 3: High Level Diagram of Ontology Merger

IV. RESULTS AND DISCUSSIONS

A) Natural Language Understanding Engine

Input: Paragraph

Earlier today, Apple unveiled two iPhone 6 models at its Special Event in Flint Center, Cupertino. The first device, the iPhone 6 comes with a 4.7-inch display and the second, the iPhone 6 Plus, comes with a 5.5-inch display. The iPhone 6 Plus has a 5.5-inch Retina HD display, with a pixel density of 401ppi and a resolution of 1920 x 1080 pixels. But there's a lot more to the iPhone 6 Plus than just its screen. Here's the full spec sheet of the iPhone 6 Plus.

Table 1: Identified Entities and Features

Entities identified by human	Apple, iPhone 6, Flint Center, Cupertino, device, 4.7 inch display, iPhone 6 Plus, 5.5 inch display, Retina HD display, pixel density, resolution, pixels, screen, full spec sheet
Entities identified by NLU engine	display, iPhone 6, models, device, pixel density, resolution, Retina HD display, Apple, models, spec sheet, Cupertino, Flint Center
Features identified by human brain	4.7 inch display, 5.5 inch display, Retina HD display, pixel density
Features identified by NLU engine	Pixel Density, Retina HD Display
Identified noun combinations	today,Apple,iPhone,models,Event,Flint Center,Cupertino.,device,inch display,Plus,inch display.,inch Retina HD display,pixel density,401ppi,resolution,pixels.,lot,screen.,spec sheet

According to an evaluation conducted on 15 people with each being provided with 5 paragraphs which adds up to 75 test cases, we have obtained the following results.

Table 2: Total Entities and Features Identified

Total Entities identified by human (EIH)	1289
Total Features identified by human (FIH)	297
Total Entities identified by NLU engine (EIN)	994
Total Feature Identified by NLU engine (FIN)	190
Correct Entities Identified by NLU Engine Comparing to human(CEIN)	872
Correct Features Identified by NLU Engine Comparing to human(CFIN)	176

Overall Accuracy comparing to human:

(CEIN+CFIN / EIH+FIH)*100

= (872+176/1289+297)*100

= 66.06%

B) JSON To OWL Mapper


```
{
  "adminStaff": {
      "departmentHead": {
        "name": "Steve Smith",
        "email": "head@university.com",
        "contact": "000-111111111"
      },
      "MarketingManager":{
       "name": "Joe Root",
        "email": "marketing@university.com",
        "contact": "000-222222222"
      }
    },
    "lecturers":[
      {
        "name": "John Smith",
        "email": "john@university.com",
        "contact": "000-3453453"
       },
       {
        "name": "Jane Doe",
        "email": "112121212121",
        "contact": "000-12345678"
       }
    ]
}
```


Figure 4: Sample JSON

Table 3: Identified classes and triples

Semantic Classes Identified	2 Classes. University, Person
RDF Triples Identified	1 Triple. University -has-Person

In the above example, a JSON object which illustrates some information about a university is taken as the input. The keys, *departmentHead*, *MarketingManager*, and the array of *lecturers* are all semantically identified as people, which is one class and instances are created in the ontology for that class with relationships each person here has with relevant to the key in which they were defined. We have used datasets provided by Yelp [31], which has provided user reviews, user information and user check in information. We have successfully mapped the given datasets into ontologies and following table shows a summary of the results we acquired.

Table 4: Results for Yelp dataset

Total JSON objects found	23,694
Total structurally different Key-Value pairs found	18
Semantic Classes Identified	8
RDF Triples Identified	6

C) Ontology Merger

Table 5: Sample Ontology Input

Ontology 1 - Triples	**Ontology 2 - Triples**
John - Has - Address John - Has - Email John - Has - Gender Sarah - Spouse- John	John - Has - Contact John - Has - Age John - Has - Languages John - Has - Religion Sarah - Has - Kids

When the above triples of the sample ontologies are merged, one single ontology is given as the output with an instance where, *`John`* is the subject and *Address, Email, Gender, Contact, Age, Languages* and *Religion* being predicates to it. Another instance, *`Sarah`* is also created with *`John`* being the spouse and having *Kids* as a predicate. Therefore, in the resulting ontology, a link between *'John'* and *Kids* can be identified. We have used the ontologies generated by the Ontology Merger mentioned in this document using the Yelp dataset and have been able to successfully merge user reviews and user information and user check-ins into one single ontology.

V. CONCLUSION AND FUTURE WORK

Currently the NLU engine is identifying noun combinations in order to identify the entities in addition to the entity identification done by the Google NLP since it has limitations of identifying some of the entities in real world. In the current process of identification of noun combinations, it uses only the sequences of nouns(NN). Therefore, as a future work, there will be more language patterns like adjective (ADJ) – noun (NN), noun (NN) – verb (VERB), pronoun (PRON) – noun (NN) combinations in order to increase the accuracy of identifying the entities and features. The platform that has been developed would enable automated knowledge understanding and integration into a generic ontology model, which could then be used to develop domain specific product by means of simply giving unstructured textual data as the raw input. The approach to this project involves coming up with a simulation of Natural Language Processing and Understanding, which is similar to humans, by using an ontological approach and representing the extracted data into a knowledge base which could later be retrieved in an automated way. Main advantage of this platform is to make developers with less knowledge of data science to benefit from using the tools provided.

VI. ACKNOWLEDGEMENT

We would like to thank our university Sri Lanka Institute of Information Technology (SLIIT), for giving this opportunity to do a research project, which pave the path to expose our skills and abilities.

REFERENCES

[1] 'Unstructured Data' [On-line]. Available: https://en.wikipedia.org/wiki/Unstructured_data [Accessed: 6-October-2017].

[2] 'Words API' [On-line]. Available: https://www.wordsapi.com [Accessed: 6-October-2017].

[3] 'Ontologies', [On-line] .Available: https://www.obitko.com/tutorials/ontologies-semantic-web/ontologies.html [Accessed: 10-March-2017].

[4] Atika Mustafa, Ali Akbar, Ahmer Sultan, Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization, International Journal of Multimedia and Ubiquitous Engineering.

[5] Text Mining Application Programming by Manu Konchadi. Published by Charles River Media. ISBN: 1584504609

[6] Yuangang Yao , Runpu Wu, Hui Liu, JTOWL: A JSON to OWL Convertor ,(2014)

[7] Automated Concept Extraction From Plain Text. Boris, GARAGe Michigan State.

[8] Wimalasuriya D.C., Dou D., Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science, 36, (2010), No. 3, 306

[9] Wimalasuriya D.C., Dou D., Components for Information Extraction: Ontology-Based Information Extractors and Generic Platforms, CIKM'10, Canada, (2010)

[10] Bontcheva K., Tablan V., Maynard D., Cunningham H., Evolving GATE to meet new challenges in language engineering, Natural Language Engineering, 10, (2004), 349-373

[11] Ferrucci D, Lally A (2004) UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering, 10 (3- 4), 327–348.

[12] Drozdzynski W., Becker M., Krieger H.-U., Piskorski J., Schäfer U., Xu F., SProUT (Shallow Processing with Unification and Typed Feature Structures) (2002), [On-line]. Available: http://sprout.dfki.de [Accessed: 10-March-2017].

[13] Buitelaar P., Siegel M., Ontology-based Information Extraction with SOBA. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, Italy, (2006).

[14] 'Ontology Based Information Extraction', [On-line]. Available: https://www.cs.uoregon.edu/Reports/ORAL-200903-Wimalasuriya.pdf [Accessed: 10-March-2017].

[15] Craig A. Knoblock , Pedro Szekely , Jose Luis Ambite , Aman Goel , Shubham Gupta, Kristina Lerman , Maria Muslea , Mohsen Taheriyan , and Parag Mallick. Semi-Automatically Mapping Structured Sources into the Semantic Web (2012).

[16] 'Steiner Tree Problem', [On-line]. Available: https://en.wikipedia.org/wiki/Steiner_tree_problem [Accessed: 10-March-2017].

[17] An Approach to Ontology Integration for Ontology Reuse, Conference Paper · July 2016 [On-line]. Available: https://www.researchgate.net/publication/307546407 [Accessed: 10-March-2017].

[18] Fisnik Dalipi, Florim Idrizi, Eip Rufati, Florin Asani. On Integration of Ontologies into E-learning Systems. In: 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks.

[19] Lijun Tang and Xu Chen, "Ontology-Based Semantic Retrieval for Education Management Systems" in Journal

of Computing and Information Technology - CIT 23, 2015, 3, 255–267 (2015)

[20] Aliyu Rufai Yauri, Rabiah Abdul Kadir, Azreen Azman, Masrah Azrifah and Azmi Murad, “Ontology Semantic Approach to Extraction of knowledge from Holy Quran”, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia IEEE Publications.(2012).

[21] Albert Comelli, Luca Agnello, Salvatore Vitabile, “An Ontology-Based Retrieval System for Mammographic Reports” in 20th IEEE Symposium on Computers and Communication (ISCC) (2015).

[22] S. R. R. Raghu A, "Ontology guided information extraction from unstructured text," International Journal of Web & Semantic Technology, vol. 4, no. 1, p. p19, 2013.

[23] M. Young, ‘CoreNLP’, [On-line]. Available: http://stanfordnlp.github.io/CoreNLP/coref.html [Accessed: 10-March-2017].

[24] ‘Ontology’, [On-line]. Available: https://jena.apache.org/documentation/ontology/ [Accessed: 10-March-2017].

[25] ‘Top Level Categories’, [On-line]. Available: http://www.jfsowa.com/ontology/toplevel.htm [Accessed: 10-March-2017].

[26] M. Keet, “Aspects of Ontology Integration”, 2004.

[27] Computational Intelligence in Data Mining - Volume 3

[28] ‘Ontology based data integration’, [On-line]. Available: https://en.wikipedia.org/wiki/Ontology-based_data_integration [Accessed: 10-March-2017]

[29] ‘Introducing JSON’, [On-line]. Available: http://www.json.org [Accessed: 10-October-2017]

[30] ‘Google NLP API’, [On-line]. Available: https://cloud.google.com/natural-language/ [Accessed: 10-October-2017]

[31] ‘Yelp Dataset’, [On-line]. Available: https://www.yelp.com/dataset/download [Accessed: 10-October-2017]

[32] ‘Probase’, [On-line]. Available: https://www.microsoft.com/en-us/research/project/probase/