

Algorithm to Match Ontologies on the Semantic Web

Alaa Qassim Al-Namiy
School of Science, Aston University
Oakville, Canada

Abstract—It has been recognized that semantic data and knowledge extraction will significantly improve the capability of natural language interfaces to the semantic search engine. Semantic Web technology offers a vast scale of sharing and integration of distributed data sources by combining information easily. This will enable the user to find the information easily and efficiently.

In this paper, we will explore some issues of developing algorithms for the Semantic Web. The first one to build the semantic contextual meaning by scanning the text, extract knowledge and automatically infer the meaning of the information from text that contains the search words in any sentence and correlate with hierarchical classes defined in the Ontology as a result of input resources. The second to discover the hierarchical relationships among terms (*i.e.* discover the semantic relations across hierarchical classifications). The proposed algorithm will be relying on a number of resources including Ontology and WordNet.

Keywords—*Semantic Web; Ontology matching; WordNet; Information retrieval; web service*

I. INTRODUCTION

There are many different design methodologies for software development, each having several advantages and disadvantages. To determine the best suited methodology for this research, an analysis was performed based on research into the various design methodologies. From the results of the analysis, the methodology chosen was the Object Oriented Design (OOD) methodology [2].

This methodology provides a number of benefits. Firstly the OOD methodology takes a real world view a system and models it using objects. This provides a natural decomposition of a system into modules. In the OOD methodology, the analysis and design phases are closely coupled together which helps in developing a prototype of the problem domain a lot quicker compared to more traditional design approaches. The reason for this is that the initial foundation of the design phase will be the information developed in the analysis phase.

The object oriented approach focuses more on data specifications, including the relationships between objects. One of the most important parts of the automatic annotation

tool is data and so by using a design methodology that is strongly focused on data, it is hoped that there will be a greater chance of developing a successful design. In addition designs created from OOD approaches map directly into implementations using object oriented programming languages such as Java or C#. Henderson-Sellers & Edwards (1990) believe that more flexible can be provided to a system based on object representation as modifications at the implementation level did not require any changes to the systems design itself due to the easily accomplished [2].

However, the traditional model for software development and the Object Oriented approaches are both suffered from the lack of identifying the role of Human Computer Interaction (HCI). The automatic annotation system clearly needs to provide interactivity for the user through a simple interface and so a significant part of the design will need to be focused on this aspect [11].

In addition, this paper describes how the methods of the system is developed for (i) a general algorithm to build the semantic contextual meaning by scanning the text, extract entity or knowledge and correlate with hierarchical classes defined in the Ontology as a result of input resources (ii) a specific algorithm to discover the hierarchical relationships among terms (*i.e.* discover the semantic relations across hierarchical classifications). The algorithms will be relying on a number of resources including Ontology and WordNet [9].

Knowledge Extraction can be used to automatically extract specific information from documents and this information could then be utilized to generate possible semantic annotations [10]. During the research stage of the author became aware of a powerful of WordNet component and Jena component which represent the foundation of the proposed system. This system was chosen to provide the Knowledge Extraction capabilities needed as it offered all of the features required including named entity recognition [4]. Another benefit of choosing WordNet and Jena components was that it has been designed to be easily incorporated into other applications and also there is a lot of documentation available explaining how to use it [12]. Figure 1 shows the layout of the main components of the system and also the input and output data.

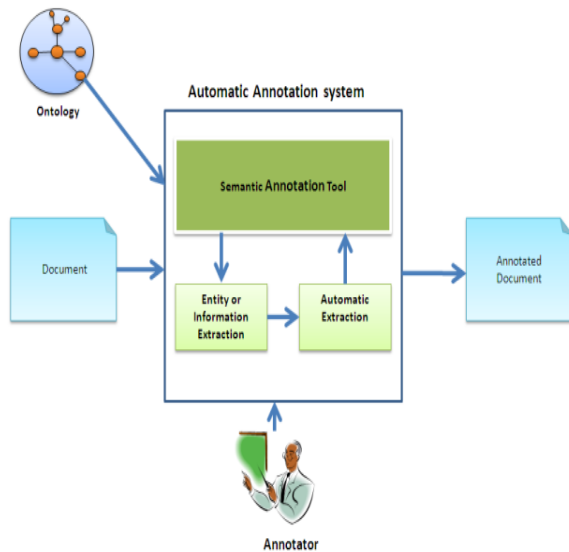


Fig.1. High level architecture of the system

The explanation of these components shown in figure 1 is as follows:

1) *Semantic Annotation Tool*

This component is the foundation of the system. It provides all the functionality required to create annotations automatically. This will include viewing Ontologies and browsing web pages. This component developed as part of this research.

2) *Knowledge Extraction*

The knowledge Extraction component of the system will analyze web pages and extract specific information found within the text. This component developed using features provided by WordNet.

3) *Automatic Annotator*

This is the main component of the system. It will take the information extracted by the knowledge Extraction component and use it to generate possible annotations. These annotations will be presented to the user through a graphical user interface. This interface will also allow the user to create annotations in RDF and save annotations to a file.

In the remainder of this paper we will first talk about the methodology and approach in Sec. II and then about the Senses Algorithm in Sec. II. In Sec. IV we will present how to build the Semantic Contextual Meaning. In sec. V the process design is discussed. Finally, we conclude our work

II. METHODOLOGY AND APPROACH

To move toward Semantic Coordination we proposed methodology based on the insight with the aim of managing conceptual structure and managing structures tag.

The proposed system gives the ability to use the semantic organization implicit with complex level in the client way that uses the language taken by the tag. In this work, three different

levels of semantic information are considered (*i.e.* knowledge) which are required to annotate structures tagged semantically with language:

- **Lexical data (Knowledge):**

Information on certain words (*i.e.* concepts) used in the tags and the relationship among them such as the word 'right' can give a meaning of correct or opposite to left.

Structural data (Knowledge): deriving information from the tags which are given in a certain structure. For example, the Entity London can be used to classify city, and London name (people with the name London).

Domain: information concerning the relation among senses of tags in a specific domain. The sense term refers to meaning of a word in WordNet *i.e.* London represent both cities in South Western Ontario, Canada and a capital city of United King Dom.

In this methodology we consider the hierarchical classifications method used for classifying particular text. The proposed algorithms must be clearly enriched with particular structure and principle functions to determine the semantic relation between the entities. This will improve the current information retrieval search anomalies.

Ontologies also play an important role in research on computational linguistics, particularly for information extraction (IE) and natural language processing (NLP) [9]. In this application area, ontologies are used as knowledge bases which provide background information for machine processing of texts. They are usually not bound to a certain domain but capture universal knowledge, and thus resemble upper ontologies in this respect. Yet their focus is less on representing the essence of the world but on capturing linguistic behavior and lexical surroundings of concepts [7]. The structure of linguistic ontologies may differ from the typical structure of concepts, instances, relations and axioms as discussed for ontologies; although they typically use hyponymic structures as a backbone and enrich them with additional concept relations. Single linguistic ontologies have recently also been transferred to the OWL format.

Used in combination with information extraction systems, linguistic ontologies can be applied in order to gather factual data for the semi-automatic construction of other (domain) ontologies [7]. They may further be used as synonym collections and dictionaries or as a major mapping reference vocabulary. Some projects also focus on supporting machine translation.

Let us look at a further example, using the three semantic levels mentioned above, to determine the semantic relations tagged London as a capital city of England, and between the entity (*i.e.* node) London city in Ontario, Canada, we can account the relations are different. Consider the mapping properties between the entities London England and London Ontario.

The lexical data notifies that the sense of the two tags is similar and refers to city. Domain notifies, among other things, that London is a city in South-western Ontario region. Finally,

structural data derived the properties of the entities and intends meaning of the node. For example the structure knowledge of "London" England will refer to all possible knowledge like: History, Middle Ages, Early modern, Local Government, Geography, Economy, Tourism, etc. while the structure of "London" Ontario may derive the following knowledge: Residents, Business, city Hall, City Life, E-Services, etc.

Example

The example shown in Figure 2 explains the category of the different levels of semantic information.

```
</owl:Ontology>
<owl:Classrdf:ID="Country">
<rdfs:subClassOf>
<owl:Classrdf:about="#UnitedKingdom"/>
</rdfs:subClassOf>
<rdfs:comment> United Kingdom of Great Britain
and Northern Ireland </rdfs:comment>
</owl:Class>
  <owl:ObjectPropertyrdf:ID=" London
">
  <rdfs:domainrdf:resource="#Capital
City England"/>
  <rdfs:rangerdf:resource="#London
England"/>
  <owl:Classrdf:ID=" History " />
  <owl:Classrdf:ID=" Middle Ages " />
  <owl:Classrdf:ID=" Local Government " />
  <owl:Classrdf:ID=" Geography " />
  <owl:Classrdf:ID=" Economy " />
  <owl:Classrdf:ID=" Tourism " />
</owl:ObjectProperty>
```

Fig.2. Example of different levels of semantic information

While structure data (knowledge) of "London", Canada could be represented as in figure 3.

The conclusion from above that the most important step is to find the word description, using WordNet which provides complex word descriptions.

III. SENSES ALGORITHM

As mentioned, to produce automatic annotation, the first phase is to create an algorithm that obtains a description about text automatically [8]. The algorithm will consider the description of a word as a textual definition, more general terms, more specific terms, or a definition in a specific language or domain. The primary source is the web text where the aim is to divide the text into sentences which enables the extraction process step by step.

To obtain the sense set for Ontology, the entire synonym words presenting in the text which are related to the semantics of the Ontology (i.e. concept)

```
<owl:Ontologyrdf:about="">
<owl:versionInfo>Version
0.1</owl:versionInfo>
<rdfs:comment>Countries Ontology,
</rdfs:comment>
</owl:Ontology>
<owl:Classrdf:ID="Country">
<rdfs:subClassOf>
<owl:Classrdf:about="#Canada"/>
</rdfs:subClassOf>
<owl:disjointWith>
<owl:Classrdf:about="#ProvincesandTerritories
"/>
</owl:disjointWith>
<owl:Classrdf:about="#AssociatedState"/>
</owl:disjointWith>
<rdfs:comment>No government body
exists</rdfs:comment>
</owl:Class>
<owl:ObjectPropertyrdf:ID=" Southwestern
Ontario ">
  <rdfs:domainrdf:resource="#London
"/>
  <rdfs:rangerdf:resource="#London Canada
"/>
  <owl:Classrdf:ID=" Residents " />
  <owl:Classrdf:ID=" Business " />
  <owl:Classrdf:ID=" city Hall " />
  <owl:Classrdf:ID=" City Life " />
  <owl:Classrdf:ID=" E-Services " />
</owl:ObjectProperty>
```

Fig.3. Example of structure data (knowledge)

will be extracted via the key functional requirements of WordNet. These main carriers of information will be analyzed based on lexical resource using the WordNet. The procedure below describes the system process manually of what WordNet needs to do, what synsets might be derived, and then how it will be annotated; our separate processes involved, namely:

A. Preliminary Analysis Of The Input Source,

This process will be the foundation phase of the system. It should provide all of the functionality required to create annotation, this will include:

- 1) ESA system access web documents to allocate text sentence.
- 2) ESA system access domain knowledge, Ontology document and WordNet interface.
- 3) ESA system establishes characteristics of the sentence, such as case, number, time and gender, etc.

B. Analysis Of The Sentence Structure,

4) ESA system access Ontology and obtain related knowledge on syntax

5) ESA implements syntactical analysis, in order to classify the syntactical structure of the paragraph. The entities and how they are arranged within the text will be analyze and describe via synset methods.

6) ESA system parse the paragraph and sentences based on the syntactic patterns derived from the WordNet dataset. Prior to linguistic processing the extraction process will divide the web text into sentences; the tokenization take place once the text is loaded, then tokenizer split a web text into textual tokens.

7) The internal structure level will be analyze and describe the relations among entities via tokenization process and pointer method ;

8) At the sentence level, characteristics sentence structures will be analyze and describe via the scanner and lexer generator methods.

9) ESA system will declare synonyms using annotation property, for example

```
<owl:AnnotationPropertyrdf:ID="synonyms" >  
<rdf:typerdf:resource="http://www.w3.org/  
2002/07/owl#DatatypeProperty"/>  
</owl:AnnotationProperty>
```

From this process the text can be converted into a view tree after interpreting the structure of the text [1].

C. Analysis of semantic, analyzing a semantic is a method of describing syntactic structure of sentences, clauses and phrases.

10) ESA system will access Ontology and obtain related knowledge on semantics

11) ESA system retrieves possible meanings of words and their proprieties from Ontology that generate a semantic descriptor of the sentence.

12) Entities will be represented as object using string of characters to identify the resource known as Uniform Resource Identifier (URIs). The advantage of using URI representation is that the object can be used to create inters relation. The word entity refer to the class of things and sub entity refer to sub class, for example "University" likely an entity.

13) ESA system will be able to indicate subclassOf relations between entities, such as department and people. This will allow detecting connections, for example employees in a specific department.

14) Semantic analysis of the entities will produce triples with two nodes. One to represent relations and terms and the

other to represent the relation which describes the sort of connections among the nodes.

15) The triples produced can be nested – which means they can further be utilized as node of the further triples. This allows for the facilitation of complex interrelations representations where and each different type of relation will describe a different linguistic phrases or keywords in text. When the triples are combined into the knowledge base, the knowledge from the texts is also transferred resulting in the generation of a more refined structure.

16) Representing knowledge can be expressed by subTypeOf and subClassOfproperty which is relevant to RDF property Entity relationships are expressed by instanceOfrelations.

17) ESA system will identify all the necessary classification of entity which is required to determine whether the entity belongs to the main class or subclass. For example, to extract entities from the instance sentence "ASTON UNIVERSITY in the heart of BIRMINGHAM CITY" would refer to: ORGANIZATION situated in LOCATION, the outcome of extraction will be, "Aston University in Birmingham City". This process is generating an automatic annotation which can be stored in a separate RDF or XML document. Furthermore, annotation properties will be used to represent the category of annotation properties in an Ontology language.

18) The terms and concepts which have been extracted will be annotated since the ontology of original text is rolled bythem. The relations can be mapped in finer way and preserved due to using the ontology by the authors. It can also offer different levels of information details by creating an overview of the contents.

19) The word relations and symbol will be schematics at the bottom will be complete.

D. Incorporation the acquired results (i.e. relevant information).The information will then be built-in as a knowledge which is consisted of terms, concepts, and their interrelations. The Jena framework will be used to manage and store the information [4]. Finally, the system will provide significant support for the agent server which has responsibility for retrieving data. Figure 43 explains the system definition phases for the proposed system.

IV. BUILDING THE SEMANTIC CONTEXTUAL MEANING

The core task of this sub-section is to introduce the independent part of the algorithm called EXTRACT–CONCEPT. The basis of this algorithm is to match the same topic in the sentence with entity / sub entity in the Ontology document

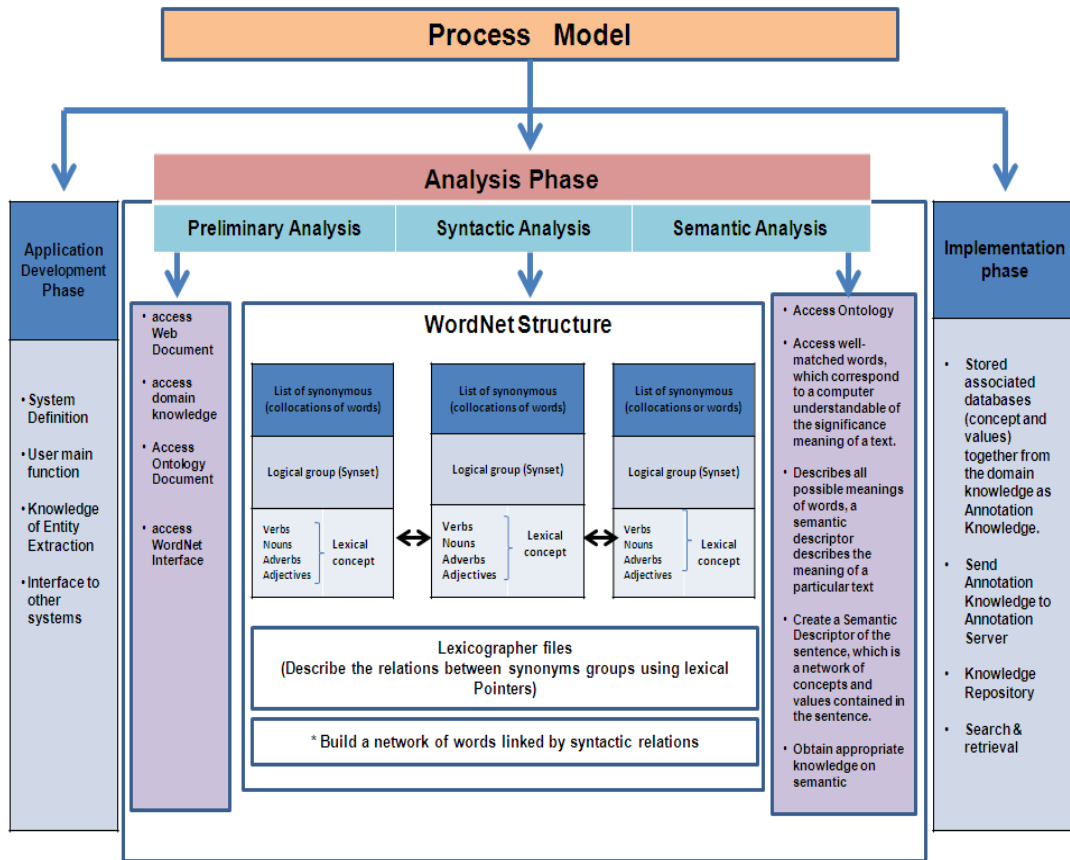


Fig.4. System Definition Phase Based Semantic Web

This algorithm designed to build the hierarchical relationships between words among concepts since the key elements of the knowledge is composed specifically an internal structure *IST*, an Ontology *GO* and a lexicon *LO*. The key input source of the algorithm is the context. The context will be analyzed based on lexical resource using the WordNet methods and will be enriched with specific functions required to scan the text, extracts knowledge and building semantic contextual meaning. The concepts expressed by a generic entity *e* (*i.e.* node) in a general text to match the hierarchical classifications [3]. In addition, the algorithm consists of functions that are necessary for retrieving information in terms of supporting user query.

In conclude the output will be the semantic relations among entity *e* and all terms belonging to internal structure. This relation represented as logical formulas (σ, ψ) that symbolize the relation among the individual concepts represented by entity and the other generic words in internal structure.

We observe that the use of WordNet as major resources allowed providing structured information relating to semantic relations among words. WordNet employ as Lexical Ontology that contains word and world knowledge which considers analyzing the text as in linguistics. The algorithm has many steps. Line 1 verifies the focus of entity *e* as a preliminary

analysis of the input. The entities and how they are arranged within the text will be analyze and describe via synset methods [5]. This step is useful to extract the meaning of entity *e* as it determines whether the entity *e* in structure *IST* exists in structure *SUBF*.

Lines 2 and 3 extract the sentence related to each entity in structure *SUBF* and provide a link between each entity for the synsets found in the Lexicon. To obtain the sense set, the entire synonym words existing in the text which are related to the semantics of the Ontology (*i.e.* concepts) will be extracted via the key functional requirements of WordNet. Consequently, in support of the example mentioned the word 'London' in WordNet has two different meanings 'Capital city of United Kingdom' and 'City of Ontario, Canada'. These senses can be recorded by the array *SynSet*, so that, *SynSet* will have two different meanings as above. The array of senses *SynSet* must provide more attention before starting analysis of the corpus of the algorithm. The set of synonyms represented by synset is a collection of senses, such that concepts can be represented by expressions which use synset in a lexicon. In Lines 4 and 5 passes through a filter out the non-relevant concept associated to generic word in internal structure.

A formula approximating the meaning can assist in building the function *INDIVIDUAL-CONCEPT* to be expressed by entity *e*. The defined classes in the ontology will

be correlated with the entity e . The function returns the relationships between the Ontology concepts, entities and objects. The combination of both domain in input to the function and the linguistic interpretation with structural knowledge (T) can assist in doing this. Finally, build the formula (σ, ψ) which represents the relation among the individual concepts represented by entity and the *local relevant axioms* as shown in Fig.5.

Algorithm 1 EXTRACT-CONCEPT (EXC, e)

Context $EXC = (IST, LO, GO)$, where IST is a internal structure

LO is a lexical Ontology – Lexicon

GO is a General Ontology Concepts - Ontology

Sentence finite list of words

entity e is a Generic Word

Variable Identifications

Sentence SynSet[][] SynsetArray of synonyms

Structure $SUBF$

Formula σ, ψ

- 1 FIND- CONCENTRATE (e, IST) $\rightarrow SUBF$;
theconcentrate $SUBF$ is a substructure of IST
- 2 **for** each entity i in $SUBF$ **do**
- 3 EXTRACT-CONCEPT(i, LO) $\rightarrow SynSet[i]$;
extract the sentence related to each entity in the structure $SUBF$
- 4 **for** each entity i in $SUBF$ **do**
- 5 NON-RELEVANT-CONCEPT($SUBF, GO, SynSet, i$) $\rightarrow SynSet[i]$;
difficult to deal with sentence are redundant
- 6 INDIVIDUAL-CONCEPT ($e, SynSet, SUBF, GO$) $\rightarrow \sigma$;
- 7 EXTRACT- RELEVANT-KNOWLEDGE ($SUBF, SynSet, GO$) $\rightarrow \psi$;
- 8 **Return** (σ, ψ) ;

EXTRACT-CONCEPT ALGORITHM

V. PROCESS DESIGN

The automatic annotation system will allow an annotator to create new annotations for a specific web page automatically by using Knowledge Extraction techniques to generate possible annotations.

A. Proposed Implementation of the Algorithm

In this stage, the most important step is to find the word description, using WordNet which provides complex word descriptions that infer meaning thus minimizing any ambiguity. In this work the description of a word can be a textual definition, more general terms, more specific terms, or a definition in a specific language or domain.

To implement the suggested algorithm Ontology needs to be created. It could be produced using the Jena Framework as Jena is able to query and store Ontology and the Jena method

of OntDocManageraddAltEntry enables [4] relationships between stored Ontologies, thus identifying the location of Ontology inside the database.

The procedure starts by reading a RDF/OWL document into a Jena model; that gives an API for handling the information [6]. Once the description of a word is recognized in the RDF document, the word available in a HTML document will be highlighted / underlined which in turn shows the description extracted from the RDF/OWL document. For example ‘Dr Tony Beaumont’ is identified in HTML page and the description of ‘Dr Tony Beaumont’ is available in RDF document as shown in Fig.6.

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:lib="http://keg.cs.aston.ac.uk/stfDtls/">
<rdf:Description about="Dr Tony Beaumont" >
<lib:creator>Alaa Al Naimy</lib:creator>
<lib:pages>10</lib:pages>
</rdf:Description>
```

Fig.5. Example of Implementation Procedure

From the above example we deduce the following should be available:

- An HTML page P with some term T of interest.
- An RDF document R which describes the term T from a set of one or more RDF descriptions.

To find T of interest in P we have to parse P that says "the term T that appears on this page will be highlighted, italics, colour, and/or larger font *i.e.* Dr Tony Beaumont ". To find R of T if that term has RDF description, then it is possible to display the properties as annotation in a new page. Once parsing the HTML page to annotate ‘Dr Tony Beaumont’, it should:

- display the description of ‘Dr Tony Beaumont’, the author and date of creation for that page in a tab or new window
- produce a new HTML page which is P with T annotated
- store specific metadata in a specific Ontology which will be achieved by adding new annotation rules.

This work suggests the usage of Jena integrated with SPARQL [14] to create a rule-based system through GeneriRuleReasoner to store the derivation data. The reason for this is to answer user's queries about the derivation of derived statements.

The system will store the derivation data in the database as the reasoner run [13]. SPARQL is a query language for getting information from RDF graphs [14]. It provides the requirements for querying by triple designs, optional patterns, disjunctions, conjunctions and supports queries like “*show me all the projects on semantic annotation*”. The *projects* will mapped by the semantic annotations. The resultant data from the project database related to the semantic annotation area will be used to identify only *projects*. The SPARQL queries results can be obtained and presented in several different forms [14].

Integrating semantic annotation within Ontology allows distinction between the same words in different contexts that give it different meanings e.g. the searching process will be easy to distinguish the word “Mississippi” the state, from “Mississippi” the river because the annotation will be with references to various concepts in the Ontology. This will improve the current information retrieval search anomalies.

This work proposes a user application which connects to an annotation server through a web site and annotates web pages of user choice. The proposed annotation server uses an Annotation engine with an embedded Jena repository, which then transfers the results of the annotation to the annotation server. The strength of this proposed work is to integrate a Knowledge Extraction platform and Ontology to provide flexibility for the formats and functions it uses. It will also support the HTML browser to display an integrated open APIs of the Ontology browser along with the documents. From looking back at the functional requirements, it is clear that the system will need to provide the following functionality:

VI. CONCLUSION

This paper provides a description of the design of the automatic annotation system. The design stage of any research is arguably the most crucial part as it describes how the system will be structured to meet the requirements. A carefully constructed design will hopefully make the system easier to implement and will minimise the number of problems encountered. Our method focuses on representing the documents succinctly and explicitly through extracting only the related resultant semantics from the document. The specific domain ontology will assist the extraction process. The guidance to the modelling process and decoupling of the knowledge base from the required documents is provided by the proposed framework.

This paper describes how the methods of the system is developed for (i) a general algorithm to build the semantic contextual meaning by scanning the text, extract entity or knowledge and correlate with hierarchical classes defined in the Ontology as a result of input resources (ii) a specific algorithm to discover the hierarchical relationships among terms (i.e. discover the semantic relations across hierarchical classifications). The algorithms will be relying on a number of resources including Ontology and WordNet.

REFERENCES

- [1] Al-Namiy, Ala'a Qasim, “Annotation Approach for Sharing Deep Web Annotations Based on Ontology”, International Conference on Internet and Multimedia Systems and Applications (IMSA 2009), August 17-19.
- [2] B. Henderson-Sellers and J. M. Edwards, “BOOK TWO of Object-Oriented Knowledge: The Working Object.”, Prentice Hall, 1994
- [3] Bojars, U., Breslin, J. G., Finn, A., & Decker, S., “Using the Semantic Web for Linking and Reusing Data Across Web 2.0 Communities”, Journal of Web Semantics, 2008, Vol.6, no.1, pp. 21– 28.
- [4] Brian McBride, (2002). Jena: A Semantic Web Toolkit. Journal of IEEE Internet Computing, Vol.6, no.6, November /December 2002, pp. 55-59.
- [5] David Milward, Marcus Bjärelund, William Hayes, Michelle Maxwell, Lisa Öberg, Nick Tilford, James Thomas, Roger Hale, Sylvia Knight, Julie Barnes, “Ontology-Based Interactive Information Extraction From Scientific Abstracts”, Journal of Comparative and Functional Genomics, Feb-Mar, 2005, Vol.6, no1-2, pp. 67-71.
- [6] Dietmar Jannach, Kostyantyn Shchekotykhin, Gerhard Friedrich, “Automated ontology instantiation from tabular web sources-The All Right system”, Web Semantics: Science, Services and Agents on the World Wide Web, September, 2009 ,Vol.7, no.3, pp.136-153.
- [7] Hepp, M., “Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies”, IEEE Internet Computing, 2007, 11(1), pp. 90– 96.
- [8] Hahn, U. & Romacker, M., (2000), Content Management in the SYNDIKATE system – How technical documents are automatically transformed to text knowledge bases”, Data & Knowledge Engineering, November 2000, Vol.35, Issue 2, , pp. 137-159.
- [9] Huang Xiao-xi and Zhou Chang-le, (2007); “An OWL-based WordNet. lexical ontology”, Journal of Zhejiang University SCIENCE A, vol.8 ,no.6, pp. 864-870.
- [10] Ishizuka, M. and Vranes S., ”A Common Concept Description of Natural Language Texts as the Foundation of Semantic Computing on the Web”, 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing 11-13 June, 2008, pp. 385 – 385.
- [11] John M. Carroll, “Conceptualizing a possible discipline of human-computer interaction”, Journal of Interacting with Computers, 2010, Vol. 22, Issue 1, pp. 3–12.
- [12] Koji Murakami, Eric Nichols, Junta Mizuno, Yotaro Watanabe, Hayato Goto, Megumi Ohki, Suguru Matsuyoshi, Kentaro Inui, Yuji Matsumoto, “Automatic Classification of Semantic Relations between Facts and Opinions”, Proceedings of the 2nd International Workshop on NLP Challenges in the Information Explosion Era (NLPiX 2010), August, Beijing, China, pp. 21-30.
- [13] Petya Osenova and Kiril Simov, “Semantic Annotation for Semi-Automatic Positioning of the Learner”, Proceedings of the 1st Workshop on Supporting eLearning with Language Resources and Semantic Data, LREC 2010, Valetta, Malta, May 2010, pp. 46-50.
- [14] Stefan Bischof, Stefan Decker, Thomas Krennwallner, and Axel Polleres, “Mapping between RDF and XML with XSPARQL”, Journal on Data Semantics, 2012, Vol.1, no.3, pp. 147-185.