An Approach To Partially Import The Ontologies On Semantic Web Based Upon User Choice

Tayybah Kiren[1], Muhammad Shoaib[1], Muhammad Tariq Pervez[2], Sonia Majid[3], Qazi Mudassar Illyas[4]

[1]Department of CS & E, University of Engineering & Technology, Lahore Pakistan
[2]Department of CS, Virtual University of Pakistan, Shadman Campus, Shadman Market, Lahore, Pakistan
[3]Lahore College for Women University, Lahore Pakistan
[4]COMSATS Institute of Information Technology, Abbottabad, Pakistan
tariq_cp@hotmail.com

Abstract: With the increase in applications using ontologies to represent semantic information, the issue of partially reusing the ontologies is getting more focus of researchers. Ontology construction from scratch is protracted and labor intensive job. Therefore, it is good to fabricate the ontologies by reusing the existing ontologies. Existing techniques for partially importing the ontology do not consider the user choice while selecting the most relevant ontologies for reusing. Most of the approaches have restriction on the size of ontology that is to be modularized. An approach for partially importing the ontologies has been presented in this paper. The proposed technique selects important keywords from a document by calculating term frequency, IR measure and precision along with class match measure to rank the most relevant ontologies. An algorithm to extract ontology fragments has been presented. This algorithm is independent of the size of ontology being reused.

[Tayybah Kiren, Muhammad Shoaib, Muhammad Tariq Pervez, Sonia Majid, Qazi Mudassar Illyas. AN Approach To Partially Import The Ontologies On Semantic Web Based Upon User Choice. Journal of American Science 2010;6(11):571-581]. (ISSN: 1545-1003).

Keywords: semantic web; ontology; partial import; knowledge management; user choice; term frequency

1. Introduction

Semantic Web is a new generation of the existing World Wide Web in which contents can be expressed in a way that their meanings are easily explicable by the search engines and it has become easier to access, share and assimilate data (Berners and Handler, 2001). Ontology is a way of formally representing a set of concepts within a domain and the relationships between those concepts. It describes the properties of that domain, and may be used to define the domain (Gruber, 1993).Ontology construction from scratch is regarded as a very protracted and labor-intensive job (Craven et al., 2000; Kietz et al., 2000; Maedche and Volz, 2001; Shamsfard and Barforoush, 2002; Khan and Luo, 2002). A better approach is to build the ontologies by reusing the existing ontologies. Many techniques have been devised so far to reuse the ontologies.

A lot of care is needed to select axioms to be copied because very small negligence in this decision can lead to information loss and even making the structure of resulting ontology awkward (Handle and Schenber, 2002). On the other hand, the technique proposed in (Bezerra et al., 2008) does not give much attention towards combining the modules to construct a partially imported ontology.

The techniques proposed in (Stuckenschmidt and Klein, 2004; Grau et al., 2005) divide all ontologies into modules. This is somehow complicated and computationally expensive task especially in cases when a large number of relevant ontologies are available. These approaches also do not give much consideration to the user's choice while selecting the relevant ontologies and relevant module.

This paper presents an approach, which has already been published in form of dissertation of the first author of this manuscript, for partially importing the ontologies. The proposed technique selects important keywords from a document by calculating Term Frequency (TF). It uses IR measure and precision along with Class Match Measure (CMM) to rank the most relevant ontologies. It also gives an algorithm to extract ontology fragments that is independent of the size of ontology being reused. Thus an ontology is constructed by partial reuse mechanism based on user choice and having no irrelevant details. Partially importing mechanism presented in the paper is very simple and having less computational complexities.

The rest of this paper is structured as follows: In section 2, the previous approaches for reusing the ontologies are described. In section 3, we give materials and methods for the proposed approach. In section 4, we present results and discussion of the proposed approach. Section 5 presents conclusions. And the last section of paper comprises of the future recommendations.

2. Related Work

In this section, we give brief overview of some available approaches to partially import ontologies.

Stuckenschmidt et al., (2004) proposed a method to reuse the ontologies by partitioning the large ontologies into small portions or parts according to their class hierarchy structure. This method firstly creates a weighted graph of the ontology to be reused and then identifies the possible partitions from this dependency graph. Difficulty with this approach is that it is only effective for very large ontologies, and this is actually ontology partitioning algorithm but not a complete methodology to reuse the ontologies (Bezerra et al., 2008).

Bezerra et al., (2008) developed a tool for extorting modules from Ontologies. Their approach was based on OOP standards like encapsulation and information hiding. The extorted module was independent and could be easily exchanged with another module with same interface as the implementation details of the module was hidden from the imported ontology. Their approach concentrates more on digging out the relevant module rather than combining the modules to make ontology. It is a promising approach but there are some deficiencies; like because of hidden implementation, the imported module might give rise to confusion at the end of person using the partially imported ontology and also their approach does not give much attention towards combining these modules to make a partially imported ontology (Grau et al., 2005).

Grau et al., (2009) introduced a module extraction approach based on logic. This approach allows extracting the modules that are based on locality. They proposed an algorithm to build a locality based module. This algorithm divides the ontology into two parts and initializes one part explicitly as void and second portion as the whole ontology. Then it performs locality test which moves the non local axioms to the first part with respect to the given signature. It performs the same thing until all the axioms in the second module are visited. These modules had the advantage that they are small in size. But at the same time the main flaw with this approach is that relationships among classes are not imported correctly, in other words, the modules are based on syntactic positioning (Grau et al., 2009).

3. Materials and Methods

This section describes the proposed technique in detail. We have divided the proposed approach into three modules. Algorithm of each module has also been presented. Figure 1 gives the overview of the proposed technique

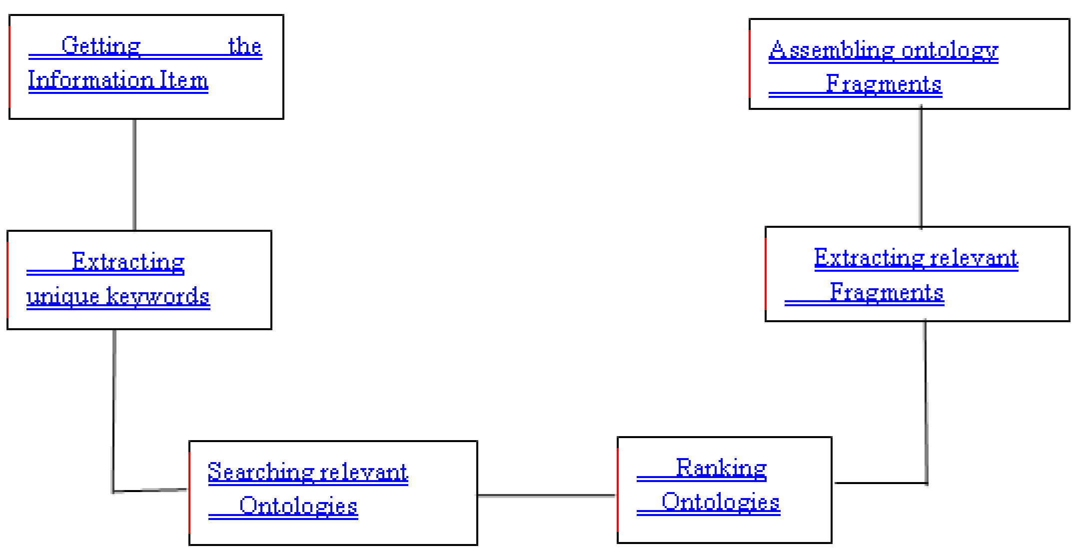

Figure 1. Graphical view of the proposed technique

The proposed technique performs its task in the following three modules.

1. Extracting keywords
2. Retrieval of relevant ontologies

3. Partially importing the ontologies

3.1. Extracting Keywords

Input: Information item
Output: list of keywords

In this module the system takes information item of user's interest. This information item may be a dynamic web page, a web page saved in a local directory or data from some online or offline repository. This document is then passed to a keyword extracting mechanism. The mechanism also sorts the important keywords from the set of extracted keywords.

The steps followed in this module are:

3.1.1. *Getting the Information Item*

In this step, any information item as per user's interest is received. The information item is received with its full completeness. This information item may be a dynamic/static web page, a web page stored in local directory or a document from an online or offline repository. The document is taken for the purpose of extracting keywords as if a user is exploring/reading some document and he/she wants to search ontology for some keyword (e .g department, faculty etc). The interest/choice of the user is given much importance because different users have different types of needs, caliber and domain of interest. The benefit of giving the choice to user to select document according to his/her interest is that the user has knowledge about the information item, extracted keywords and thus retrieval of relevant ontologies is easy and comfortable.

3.1.2. *Extracting unique keywords*

In this step, we extract unique words from the information item. In the extracted unique words the unwanted words like stop words and other signs are not included. As they are neither unique words nor of interest to the user. First the information item is converted to plain text. Then unique keywords are extracted from it and TF of each keyword is computed. The formula for calculating TF as given by (Yates and Neto, 2005)

$$tf_{i,j} = n_{i,j} / E_k n_{k,j} \qquad (1)$$

Where $n_{i,j}$ is the number of times a word t_i appears in the document p_j and the denominator in the above equation is sum of occurrences of all the words in the document. After this, the words with TF value above the threshold 0.043 are selected.

Algorithm for Extracting Keywords

The processing steps for the Module of extracting keywords are illustrated in an algorithmic form below:

ALGORITHM: Generating a set of unique keywords from an information item.
INPUT: Information item
OUTPUT: Set of keywords.
STEP1: /*getting the Information item
1.1: Get the document (information item) with its full completeness
STEP2: /* Extracting unique keywords
2.1: **IF** it is a web page
2.2: Convert the web page into UTF-8 encoding
2.3: Perform preprocessing on the page (e.g. Removing HTML tags, Multimedia Contents, links, frames etc)
2.4**: ELSE GOTO** step 2.5.
2.5: Convert the document to plain text (Removing stop words etc.)
End If
2.6: Parse the whole document in separate words
2.8 Compute number of times each word appears in the document
2.9 Compute the total number of time all the words appear in the document
2.10 Compute TF (Term frequency) of each word by dividing the results of Step 2.8 by 2.9
2.11: Save words with TF value above the threshold.
END

Algorithm of extracting keywords shows a complete process of extracting keywords from an information item. If an information item is a web page then following two steps are performed: 1) the information item is converted into a UTF-8 encoding document. 2) A preprocessing is performed in which the HTML tags, multimedia contents, links and frames etc. are removed. The purpose of performing preprocessing is to reduce the size and complexity of the information item. Then the information item is converted to a plain text. Now, in the whole information item, each word's occurrence is computed. Then, an aggregation of all word's occurrences in the information item is computed. Finally TF of each word is computed by dividing the occurrence of a single word by the total occurrences of all words in the information item. Figure 2 shows graphical representation of this algorithm.

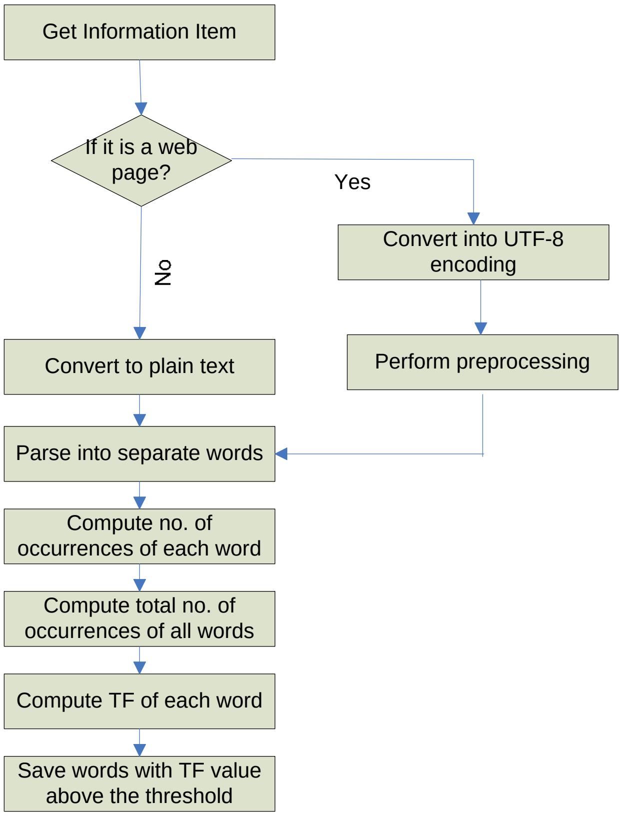

Figure. 2. Graphical view of algorithm of extracting keywords

3.2. *Retrieval of relevant ontologies*

Input: Set of keywords
Output: List of relevant ontologies

This module uses the list of keywords generated by the module of extracting keywords as an input. After retrieving the ontologies relevant to the set of keywords, this module ranks the ontologies on the basis of their similarity to the keywords set. The module performs the following two steps:

3.2.1. *Searching ontologies*

This step is concerned with giving a query consisting of keyword(s) to the ontology search engine and saving the set of relevant ontologies retrieved. Output of this step will be a set of retrieved ontologies from the Semantic Web.

3.2.2. *Ranking the Ontologies*

Input: List of ontologies
Output: Set of most relevant ontologies

This step of the proposed technique is concerned with re-Ranking of the list of ontologies retrieved using the CMM, IR measure and precision. First we find out that to which extent the class of ontology is matched to the keywords, which is determined by CMM. The formula for CMM is as given by (Alani and Brewster, 2008).

$CMM(o,K) = \alpha E(o,K) + \beta P(o,T)$

The number of classes of ontology is calculated whose labels match with the K. Then this information is used to calculate the CMM. Where α and β are the measures of exact and partial match. Now the Precision of ontologies selected is calculated. Precision is based upon the user choice, mean, if the user thinks that the document is relevant. Formula for the precision is as given by (Yates and Neto, 2005).

$P=|Ra|/|A|$

The ontologies retrieved are ranked according to the decreasing order of precision and CMM.

Algorithm of this module is divided into two parts/algorithms:

1 Algorithm for retrieving ontologies
2 Algorithm for ranking/retrieving most relevant ontologies

ALGORITHM: Retrieving Ontologies
Input: A list of keywords
Output: A list of ontologies: List_Ontologies

1. Get a list of keywords
2. Provide the list of keywords to the search engine
3. Save the relevant ontologies in List_Ontologies

Return List_Ontologies
End

ALGORITHM: Retrieving Most Relevant Ontologies
Input: A list of relevant ontologies: List_Ontologies
Output: A list of most relevant ontologies: List_MostRel Ontl
For each ontology in List_Ontologies to length (List_Ontologies) **do**
Calculate CMM of List Ontologies [ontology] as per equation 2
Apply IR measure and precision on List_Ontologies[ontology] as per equation 3
List_MostRel_Ontl[ontology]=List_Ontologies[ontology]
End For

Sort List_MostRel_Ontl in decreasing order of precision and CMM
Return List_MostRel_Ontl
End

3.3. *Partially Importing the Ontologies*

Input: Output of Module 1 and Module 2
Output: Partial ontology

This module of the proposed technique is concerned with actually making it possible to generate the ontology through partial importing from the other ontologies. It performs its task by pursuing the following steps:

3.3.1. *Extracting relevant fragments*

This step deals with the finding and retrieval of the fragments or portions of the selected ontologies that best match the set of important words. An algorithm is designed to extract relevant fragments from selected ontologies. Figure 3 shows graphical representation of the algorithm.

ALGORITHM: Generating fragments of ontologies relevant to the set of keywords.
INPUT: An ontology and set of keywords
OUTPUT: A relevant ontology fragment
1.1:Create **TobeKept-** A data structure containing nodes included in the retrieved fragment
1.2 Create **TobeChecked**-A data structure containing all the nodes in the ontology
1.3: Current=first element of TobeChecked
1.4: **WHILE not end of** TobeChecked DO
1.5: x = Current
1.6: Compare x with set of keywords
1.7: **IF** matched with any one from the set
1. 8:**IF** x ¢ TobeKept
1.9: Insert it and only its direct subclasses and properties into the TobeKept (by Referring to its child nodes)
END IF
END IF
1.10: Current=next element of the TobeChecked
END {WHILE}
Return TobeKept
END

3.3.2. *Assembling the Fragments*

This is the final step of our proposed approach where we will finally construct ontology by assembling the ontology fragments resulted from step 1 and by adding the classes and properties that are defined specifically for that ontology.

4. Results and Discussion

In this section, we validate the proposed technique and present its result. As an experiment we use a document stored in our local disk as the information item. This document is about a university which is organizing a software exhibition where PhD thesis (S/W Products) is placed. Hence, we construct ontology about the university described in the document. We show that how our proposed technique is used to construct a partially imported ontology of the university described in the document.

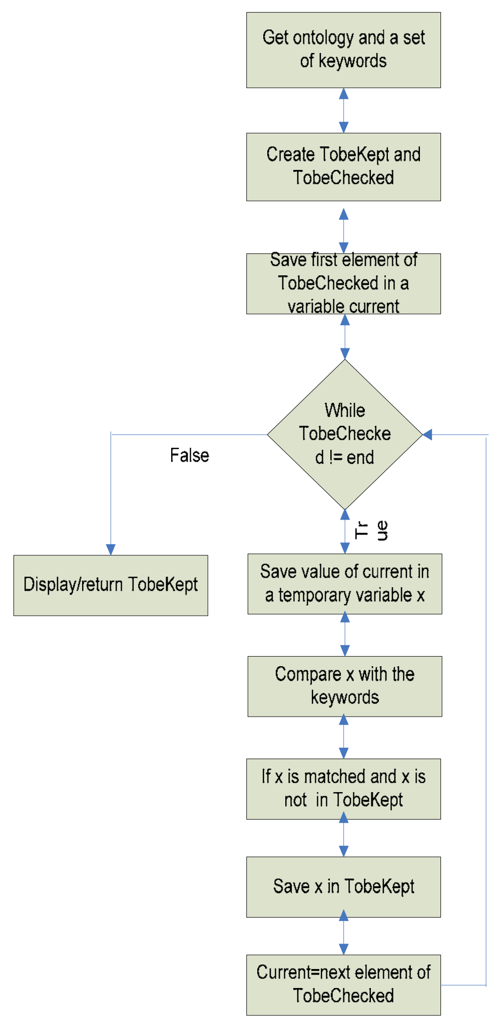

Figure 3. Flow diagram of algorithm of generating fragments of ontologies

Module 1: Extracting keywords

The document is fed into a keyword extractor (Webseo, 2010). We calculate the TF value of each keyword extracted by the keyword extractor and select the words with TF value above the threshold value 0.043. Following table shows the list of keywords and their TF values. There are a total of 25 keywords extracted.

As table 1 shows, the set of keywords having TF value greater than 0.043 is K= {University, Department, Thesis, Faculty, Dean, Magazine, Institute, Exhibition, Publisher}.

Module 2: Retrieval of relevant ontologies

Using the available list of keywords (table 1) related ontologies can be searched from the web. SWOOGLE (Swoogle, 2010) is used for this purpose. Table 2 shows the set of ontologies present on web that are considered to be relevant to the set of keywords according to the professionals.

Once ontologies are retrieved, we re-Rank and then select the most relevant ontologies from the resulting ontologies using the precision measure which is based on the user choice and the CMM value. Table 2 shows re-ranking of the ontologies.

In this experiment we use threshold value of 30% for precision and 4.0 for CMM. Only the ontologies with CMM greater than or equal to 4.0 according to (Tun and Dong, 2008), and precision greater than 30 %. are selected as shown in table 4.

Table 1: Set of keywords and their TF value

S. No	**Keywords**	**$n_{i,j}$**	**TF**
1	University	10	0.083
2	Department	9	0.074
3	Faculty	9	0.074
4	Thesis	8	0.066
5	Magazine	8	0.066
6	Dean	7	0.057
7	Institute	6	0.049
8	Publisher	7	0.057
9	Exhibition	6	0.049
10	People	5	0.041
11	School	5	0.041
12	Literary	5	0.041
13	Courses	4	0.033
14	Scientific	4	0.033
15	Physics	3	0.025
16	Statement	3	0.025
17	Information	3	0.025
18	Application	3	0.025
19	Method	3	0.025
20	Guide	3	0.025
21	Discipline	2	0.016
22	Graduate	2	0.016
23	Group	2	0.016
24	Institution	2	0.016
25	Knowledge	2	0.016

Table 2. Set of ontologies relevant to K

S. No	Ontology URLs
1	http://ebiquity.umbc.edu/ontology/contact.owl.
2	http://www.cs.toranto.edu/semanticweb/maponto/Maponto/Examples/univ-cs.owl
3	http://svn.mindswap.org/pallet/branches/dlsafe/ontology.owl
4	http://visitology.com/ont/bug/import/clean/academic.owl
5	http://owl.cs.manchester.ac.uk/2008/iswc-tones/ontologies/univ-bench.owl
6	http://swrc.ontoware.org/ontology
7	htttp://www.cs.man.ac.uk/~rector/Modules/COMP60461-2008/lab-material/Tangled-ontology-from-personnel-dept-01-01.owl
8	http://ontoworld.org/index.php/special:ExportRDF/Mike-Dean
9	http://www.mindswap.org/2004/multipointOnt/Factoredontologies/italianUniversities/it-apartition1.owl
10	htttp://iswc2006.semanticweb.org/submission/iswc2006 in use-Allemang-Dean

Table 3. Table for re-Ranking

S. No	**Set of Retrieved ontologies**	**CMM**	**Relevancy according to user**	**P**
1	http://www.cs.toranto.edu/semanticweb/maponto/Maponto/Examples/univ-cs.owl	6.2	T	100.00%
2	http://downloads.dbpedia.org/3.2/en/dbpedia-ontology.owl	3		50.00%

3	http://swrc.ontoware.org/ontology	4.4	T	66.00%
4	http://www.aktors.org/ontology/portal	2		50.00%
5	http://www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml	3.8		40.00%
6	http://www.mindswap.org/2004/multipointOnt/Factoredontologies/italianUniversities/it-apartition1.owl	4.2	T	50.00%
7	http://annotation.semanticweb.org/iswc/iswc.owl	2.4		43.00%
8	http://morpheus.cs.umbc.edu/aks1/ontosem.owl	1.8		38.00%
9	http://www.srdc.metu.edu.tr/ubl/ContextOntology/naics.owl	1		33.00%
10	http://owl.cs.manchester.ac.uk/2008/iswc-tones/ontologies/univ-bench.owl	4.2	T	40.00%
11	http://srdc.metu.edu.tr/ubl/ContextOntology/unspc.owl	0.8		36.00%
12	http://www.apps.ag-nbi.de/makna/semwebexport?language=rdf&model=inferenced			32.00%
13	http://www.cs.toranto.edu/~yuana/research/maponto/Bibliographic_Data.owl	2.8		30.00%
14	http://www.webkursi.lv/luweb05fall/resources/university.owl	4		28.00%
15	http://svn.mindswap.org/pallet/branches/dlsafe/ontology.owl	4.8	T	34.00%

Table 4. Selected ontologies

S. No	**Set of Retrieved ontologies**	**CMM**	**Relevancy according to user**	**P**
1	http://www.cs.toranto.edu/semanticweb/maponto/Maponto/Examples/univ-cs.owl	6.2	T	100.00%
2	http://swrc.ontoware.org/ontology	4.4	T	66.00%
3	http://www.mindswap.org/2004/multipointOnt/Factoredontologies/italianUniversities/it-apartition1.owl	4.2	T	50.00%
4	http://owl.cs.manchester.ac.uk/2008/iswc-tones/ontologies/univ-bench.owl	4.2	T	40.00%
5	http://svn.mindswap.org/pallet/branches/dlsafe/ontology.owl	4.8	T	34.00%

Module 3: Constructing partially imported ontology

Now, from the above five most relevant ontologies we extract the most relevant fragments using the proposed algorithm for extracting relevant fragments. For example, I use two ontologies (Maponto, 2010; Org, 2010). Ontology in (Maponto, 2010) contains the classes and relationships mentioned in the table 5 in it. This is clear that if we want to reuse this ontology it is not suitable to import the whole ontology because of space complexity. Therefore, instead of importing the whole ontology only a relevant portion of ontology is extracted using the proposed algorithm.

Table 5: Table for TobeChecked

S.No	Classes	Direct classes and properties
1	Work	Research Course, Work Title
2	Journal	
3	School	
4	Faculty	Professor, Lecturer, Post Doc, Teacher of
5	Worker	Faculty, Admin Staff, Assistant
6	Periodical	Journal, Magazine
7	Article	Book article, Journal article, Conference Paper, Technical Report
8	Thesis	Master Thesis, Doctoral Thesis
9	Professor	Full Professor, Associate Professor, Dean, Visiting Professor, Tenured
10	Admin Staff	Director, Dean, System Staff, Clinical Staff
11	University	Master Degree from, Doctoral Degree from
12	Person	Student, Member, Research Interest, email Address, Person Name, Age
13	Student	Undergraduate Student, Graduate Student, Takes course
14	Magazine	
15	Department	
16	Dean	
17	Organization	Department, School, Institute, University, Research group, Affiliated Organization
18	Research Group	
19	Research Assistant	
20	Doctoral Thesis	
21	Publication	Thesis, Book, Manual, Publication author, Publication Research, Pub Title, Periodical
22	Course	Has TAs, has Instructor

Now for extracting the classes that are partially or exactly matched with the set of keywords is done by traversing table 5. In the 1st iteration the 'Work' is not matched with any of the keyword, therefore, the algorithm again goes to step 1.4 (Algorithm of generating fragments). Now, the next element is 'Journal' which is also not matched with any keyword. This process will be repeated until table 5 is finished. Table 6 shows the contents of TobeKept.

Table 6: Table for TobeKept

S.No	Classes	Direct Subclasses and Properties
1	Faculty member	Professor, Lecturer
2	Thesis	Master Thesis, PhD Thesis, Supervised by
3	University	Has
4	Magazine	
5	Department	Part of, has
6	Dean	Head of

The graphical representation of the fragments extracted sown in table 6 is given in the figure 4.

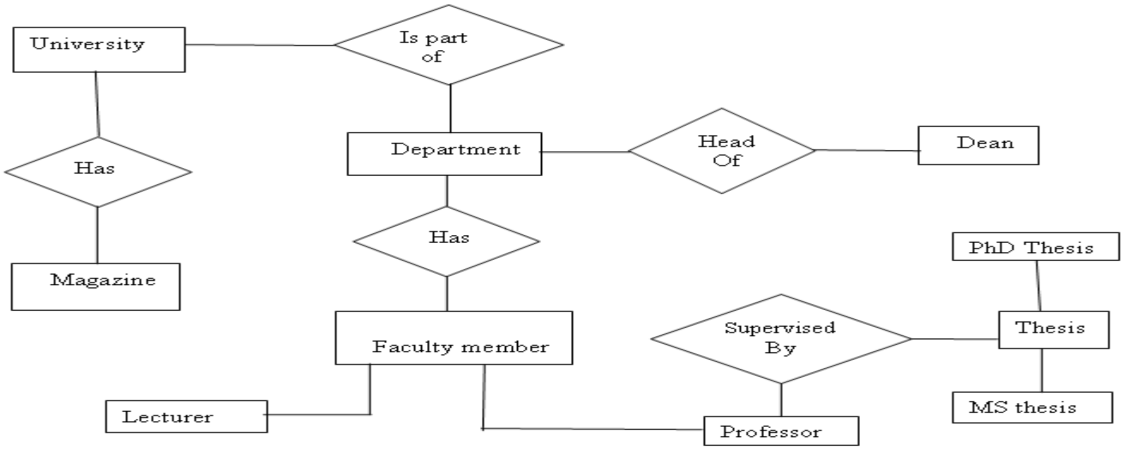

Figure 4. Graphical representation of extracted Fragments

Now if we run the algorithm on the 2nd ontology (Maponto, 2010).

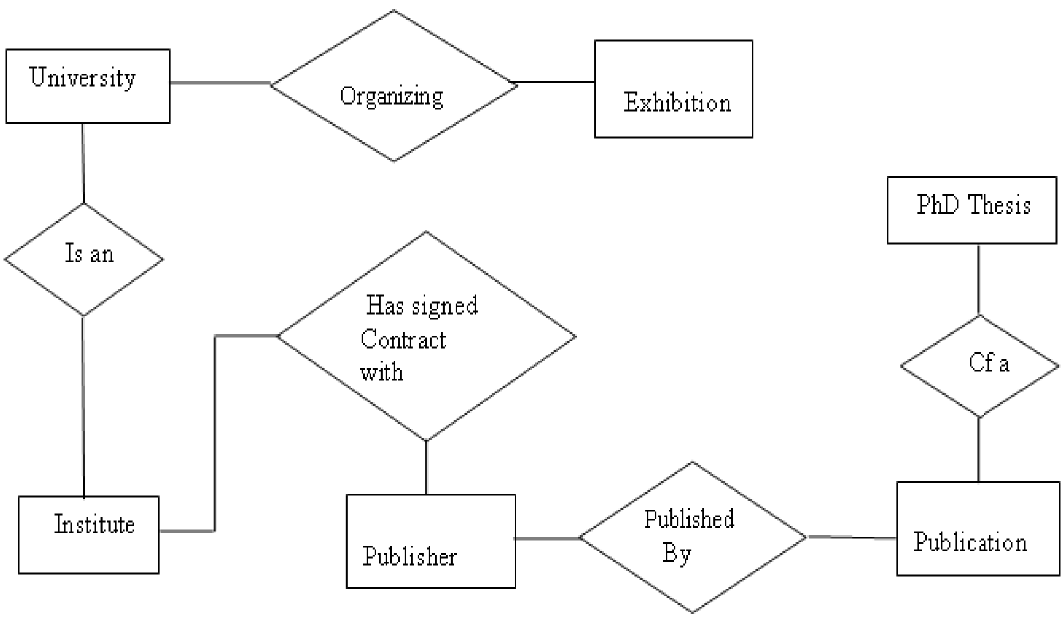

Figure. 5. Graphical representation of extracted Fragment

Table 7 shows the relations and classes of the ontology to be constructed.

Table 7: Defined classes and properties

Classes	Relations
University	Has, is a, signed contract with, Organizing
Department	Part of, has
Dean	Head of
Magazine	Published by
Faculty member	Employee of, Lecturer, Professor
Thesis	Published in, Placed in, supervised by, MS Thesis,

	PhD Thesis
Publication	Published by, of a
Publisher	Publishes
Exhibition	Organized by
Software Exhibition	Is an
S/W Product	Presented

Figure 6 shows the graphical representation of the ontology constructed by assembling the above fragments and with adding some new classes and relations.

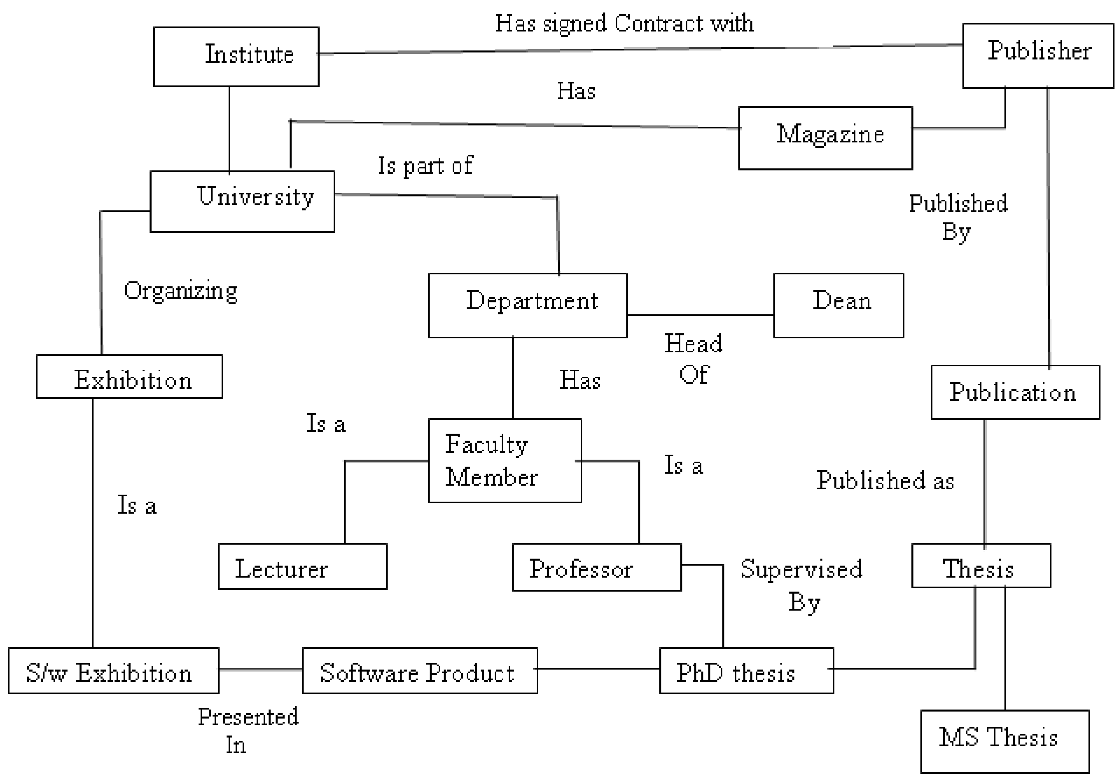

Figure. 6. Graphical representation of Resulting ontology

The resulting ontology is created by reusing the fragments extracted from the existing ontologies. It also contains some other classes and relations that are unique to it. The resulting ontology is the ontology of university that is organizing a software exhibition where the PhD thesis is published as a publication in a magazine. University has department and department has faculty member which include both lecturers and professors and a PhD thesis is supervised by a professor. Department is headed by Dean. Institute has signed a contract with a publisher and institute is a university in our example. Resulting ontology has no irrelevant details, thus reducing the large memory requirement for storing it.

5. Conclusions

This paper proposes an approach to partially import the ontologies based upon the user choice. It allows a user to built ontology of any document available at any source. This approach extracts keywords from the document of interest to user by including an important measure term frequency. Then it searches the existing ontologies from the web relevant to these keywords, ranks them and selects the most relevant ontologies with the help of precision measures and CMM, taking into account the user's choice. Then it traverses through the ontology and selects the classes which are matched to any of the set of keywords along with their direct subclasses and properties, and makes a fragment of it. Then it constructs a partially imported ontology by defining some new classes and properties according to requirement and assembling and including the fragments extracted from the existing ontology. The end result of our approach is an ontology which is constructed by partial reuse mechanism based on user choice and having no irrelevant details. Hence it is concluded that if we partially import the ontologies according to the user choice then we result with the partially imported ontology which do not contain the extra details and also the partially importing mechanism is very simple with very less computational complexities.

6. Future Recommendations

The proposed technique has not been implemented. Work can be carried out with the aim to develop a complete application that enables the user to partially import the ontologies. This approach will be applied for knowledge management and knowledge management is an asset for success and survival in an increasingly competitive and global market, so using ontologies for the knowledge management in a good way by partially importing the ontologies is the need of time and this is area where research can be carried out.

It is needed to explore more robust strategies to evaluate the quality of the resulting partially imported ontology by our approach. Fragment ranking can improve the quality of our resulting ontology. The research can be carried out in order to implement ontology fragments by fragment ranking.

Acknowledgements

We are thankful to the Higher education Commission of Pakistan for her generous financial support. This research paper is extracted from dissertation of the first author. She is very grateful to her supervisor and colleagues who helped her in extracting, writing and finalizing this research paper from her dissertation.

Corresponding Author:
Muhammad Tariq Pervez
Department of Computer Science
Virtual University of Pakistan, Shadman Campus,
Shadman Market, Lahore Pakistan
E-mail: tariq_cp@hotmail.com

References

[1] Berners L. T, Handler, J. The Semantic Web, Scientific American Magazine, 2001.
[2] Gruber T. A translation approach to portable ontology specifications, 1993.
[3] Craven M, DiPasquo D, Freitag D, McCallum A.K, Mitchell T.M, Nigam K, Slattery. Learning to construct knowledge bases from the World Wide Web, Elsevier Science Publishers Ltd, 2000.
[4] Kietz J.U, Maedche A, Volz R. A Method for Semi-Automatic Ontology Acquisition From a Corporate Intranet, 2000.
[5] Maedche A, Volz R. The Text-To-Onto Ontology Extraction and Maintenance Environment, To appear in Proceedings of the ICDM Workshop on Integrating data mining and knowledge management, San Jose, California, 2001.
[6] Shamsfard, Barforoush M.A. An introduction to Hasti: ontology learning System, Artificial intelligence and Soft computing, 2002.
[7] Khan L, Luo F. Ontology Construction for Information Selection, In Proc. of 14th IEEE International Conference on Tools with Artificial Intelligence, Washington DC, November 2002, 122-127.
[8] Handle J, Scheriber G. Web Ontology Language, 2002.
[9] Bezerra C, Freitas F, Euzenat J, Zimmermann A. An Approach for Ontology Modularization, 2008.
[10] Stuckenschmidt H, Klein M. Structure-based partitioning of large concept Hierarchies, International Semantic Web Confernce, 2004, 289-303.
[11] Grau B.C, Parsia B, Sirin E. "Modularizing OWL Ontologies," Proceedings of the 4th International Semantic Web Conference, 2005.
[12] Grau B. C, Horrocks I, Kazakov Y, Sattler U. Extracting Modules from Ontologies:A Logic-based Approach, Springer-Verlag, 2009, 159-186.
[13] Yates R.B, Neto B.R. Modern Information Retrieval, ISBN 9780321416919, 2005.
[14] Alani H, Brewster C. Metrics for Ranking Ontologies,4th Int. EON Workshop, 15th Int. World Wide Web Conf., Edinburgh 2005 of Knowledge Representation and Reasoning,2008.
[15] Webseo,P. Retrieved May 23, 2010, from http://www.webseo.com.au/keyword _extractor.html. 2010.
[16] Swoogle. Retrieved May 23, 2010, from http://swoogle.umbc.edu/, 2010.
[17] Tun N.N, Dong J.S Ontology Generation through the Fusion of Partial Reuse and Relation Extraction, in Proceedings, Eleventh International Conference on Principles, 2008.
[18] Maponto. Retrieved May 05, 2010, from http://www.cs.toronto.edu/semanticweb/maponto/, 2010.
[19] Org, O.W. Retrieved June 15, 2010, from http://swrc.ontoware.org/ontology, 2010.

8/31/2010