An Integrated Ontology Ranking Method for Enhancing Knowledge Reuse

Prof. R. Subhashini [#1], Dr. J. Akilandeswari [*2], Surumi Haris [#3]

[#] Department of Computer Science and Engineering
Sona College of Technology, Anna University, Salem, Tamil Nadu, India
[1] subhageorg@gmail.com
[3] surumiharis@gmail.com
[*] Department of Information Technology,
Sona College of Technology, Anna University, Salem, Tamil Nadu, India
[2] akila_rangabashyam@yahoo.com

Abstract—The Semantic Web is a mesh of information linked up such that it can be easily processed by machines. The focus of semantic web is to share data instead of documents and the ontologies act as the mainstay of the semantic web. Ontologies are used to represent domain knowledge in semantic web. As ontologies have many applications in various prominent fields, ontology reuse is becoming increasingly significant. To facilitate the search and reuse of ontologies many ontology libraries and search engines are in existence. But as there are a huge number of ontologies, the need for a faultless ontology ranking algorithm will help users find the best ontology that suits their purpose. The existing ranking methods focus only in the lexical level matching which does not provide exact results to the user. In order to overcome this, the paper proposes an integrated ranking approach which associates two ranking algorithms namely Content based ranking and ontology rank that encourages semantic matching.

Keyword-Ontology, Ontology Ranking, Relation matching, Taxonomy matching, Concept matching, Ranking Techniques

I. INTRODUCTION

Semantic web is an effort to enhance current web so that computers can interpret and process the information presented on the web. By including the semantic content in the web, semantic web aims at converting the current web into a web of data. In semantic web the relevant information sources are semantically structured so that the system can understand and respond to complex human requests based on their meaning. Because of this the semantic web ensures better understanding of the information and also provides a [11] good quality knowledge representation based on the user query.

Ontologies are the most important tool for knowledge representation in semantic web as they logically relate a large amount of data. They are widely used in fields like artificial intelligence, systems engineering, software engineering, biomedical informatics, information extraction, semantic web etc. [13]. As the time, effort and skill required for building ontologies remains high, the reuse of existing ontologies are widely encouraged. In order to achieve an effective level of knowledge reuse, the search engines like Swoogle, OntoKhoj which are capable of finding the relevant ontologies are currently used. The search engines can find any number of ontologies, so there is an increased need for a proper ranking method in order to rank the returned lists of ontologies in terms of their relevancy to the query. This could save a lot of time and effort by reducing the need to inspect in detail each and every ontology returned to find out the matching search term.

Ranking ontologies is an important issue, especially when many potentially-relevant ontology is found out by the search engines. Swoogle [5] and OntoKhoj [15] rank ontologies using a PageRank [14] method that analyses links and referrals between ontologies to identify the required ontologies. But, the majority of ontologies available on the Web are poorly connected, which will likely produce poor PageRank results. Another problem is that until now, much research on ontology selection and ranking is focused on lexical-level support only. However, in these cases it is almost impossible to find an ontology that includes all the concepts matched by the search terms at the semantic level. Searching an ontology that meets users' needs requires a new ontology selection and ranking mechanism based on semantic similarity matching. Similarity measures have been widely explored in information retrieval systems to provide better ranking for query results. The Semantic Similarity Measure (SSM) calculates how close the concepts of interest are laid out in the ontology structure.

The user requirement for ontology selection includes identifying an ontology that contains and well represents concepts and relations for user's application needs. Hence this paper proposes a valid ranking algorithm based on Ontology Rank [1] and Content based ranking [8]. This algorithm tries to reuse existing ontologies instead of building new ones.

The objective of this paper is to propose a framework for the ontology ranking process to retrieve and rank ontologies that fit user requirements. Background concerning ontology ranking is reviewed in the following section. A full description of the architecture and ranking method of the proposed system is given in section3. Experiments and results are described in the section 4 and conclusion is discussed in the final sections of the paper.

II. BACKGROUND

The search engines play a vital role in retrieving the information required by the user. However, the retrieved web pages also contain ineffective or irrelevant information. The latest web architecture, represented by semantic web overcomes this limitation by applying the ranking algorithms. The ranking algorithm extracts the information for the user queries from the semantic search engine and provides the desired result. Evaluating and ranking ontologies can be based on many different criteria. There are several techniques for searching and ranking ontologies. Some of the ontology ranking algorithms are Swoogle, AKTive rank, OntoRank, OntoQA etc.

Harith Alani et.al. [7] proposed AKTive Rank which is a technique for ranking ontologies based on different analytical measures. AKTiveRank applies four measures to evaluate different representational aspects of the ontology and calculate its ranking. The four measures used in AKTiveRank are Class Match Measure (CMM), Centrality Measure (CEM), Semantic Similarity Measure (SSM), and Density Measure (DEM). CMM evaluates the coverage of an ontology for given search terms, CEM measures the centrality of the given search terms in the hierarchy. SSM is a metric to measure the distance between concepts in the taxonomy, while DEM indicates how densely a given concept is defined in the ontology by summing up the number of its subclasses, super classes, siblings, and relations connecting the concept. Each ontology is examined separately and once those measures are all calculated for the particular ontology, the resulting values will be merged to produce the total rank for the ontology.

OntoRank algorithm [10] is another one promising ranking algorithm, which was developed by Maryland University. It uses the link analyse method which is similar to PageRank algorithm. The ontology is arranged by their order of importance in the OntoRank algorithm. It shows the quality of ontology and the influence of one ontology extends to another. The two concepts C1 and C2 are considered as a reference relation, if and only if a relationship exists between the two concepts in a relationship set [10] {rdf: type, rdfs: subclass, rdfs: domain, rdfs: range}. The link analyse method is not used widely as there exists only small number of links between ontologies.

Swoogle [5][6] is a crawler based indexing and retrieval system for Semantic Web that crawls and discovers documents written in RDF, OWL etc. It classifies a Semantic Web Document (SWD) as Semantic Web Ontology (SWO) which defines new terms and Semantic Web Databases (SWDB) that makes assertions about individuals. It contains about 10,000 ontologies and has ranking standards very similar to the Google search engine. Swoogle adopts a PageRank-like method to rank ontologies by analysing links and referrals between ontologies.

The Content Based Ontology Ranking algorithm was proposed by Matthew Jones and Harith Alani [8]. This downloads a list of ontologies from a search engine and based on the term given by the user or the knowledge engineer, the retrieved ontologies are ranked. The ranking is done according to the number of concept labels in those ontologies which matches a set of terms extracted from WordNet. It is done related to domain of knowledge identified by the knowledge engineer’s original search terms.

OntoQA [9] ranks populated ontologies using a set of metrics and by their relation to a set of keywords. It ranks ontologies related to a user given set of terms and it also helps to improve the metrics introduced previously and add a number of new metrics that help in better ontology evaluation. OntoQA is a popular ontology ranking method unlike other approaches since it requires minimum user involvement.

OntoSelect [3] provides the user an up-to-date overview of web accessible ontologies and it is an ontology library that is updated continuously. The ontologies are organized in a meaningful way and have automatic support for ontology selection in knowledge mark up. OntoSelect uses a crawling procedure that monitors the web for any newly published ontology in RDF/S, DAML or OWL formats. Collected ontologies are analysed using the OWL API and allows for the extraction of structure and content of any RDF/S, DAML or OWL ontology. OntoSelect library covers a wide range of topics and domains that currently has around 745 ontologies in it.An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

III.COMBINED RANKING ALGORITHM FOR RANKING ONTOLOGIES

Ontology Rank [1] cannot be applied to publicly available ontologies as the relations between concepts have not been identified precisely. Content Ranking algorithm [8] although experiments a new method of search term expansion using WordNet, works good if it is combined with a strong ranking algorithm. In view of this, the paper tries to propose a combined ranking algorithm called the Content-OR which is based on the Content Ranking algorithm and Ontology Rank..

A. Architecture of Content-OR

The architecture of combined ranking algorithm is depicted in figure 1. It is divided into four components. They are query interface, ontology reader, ranking module and an external module which includes an ontology search engine and an ontology repository. The first component query interface receives keyword based query from the user or an agent. The query contains the terms to search for. The system uses an external web search engines such as Swoogle to build a local ontology repository. The list of ontology candidates will be gathered from the external search engine and check whether it is there in the ontology repository. If it is not there, the ontology will be downloaded to the local ontology repository.

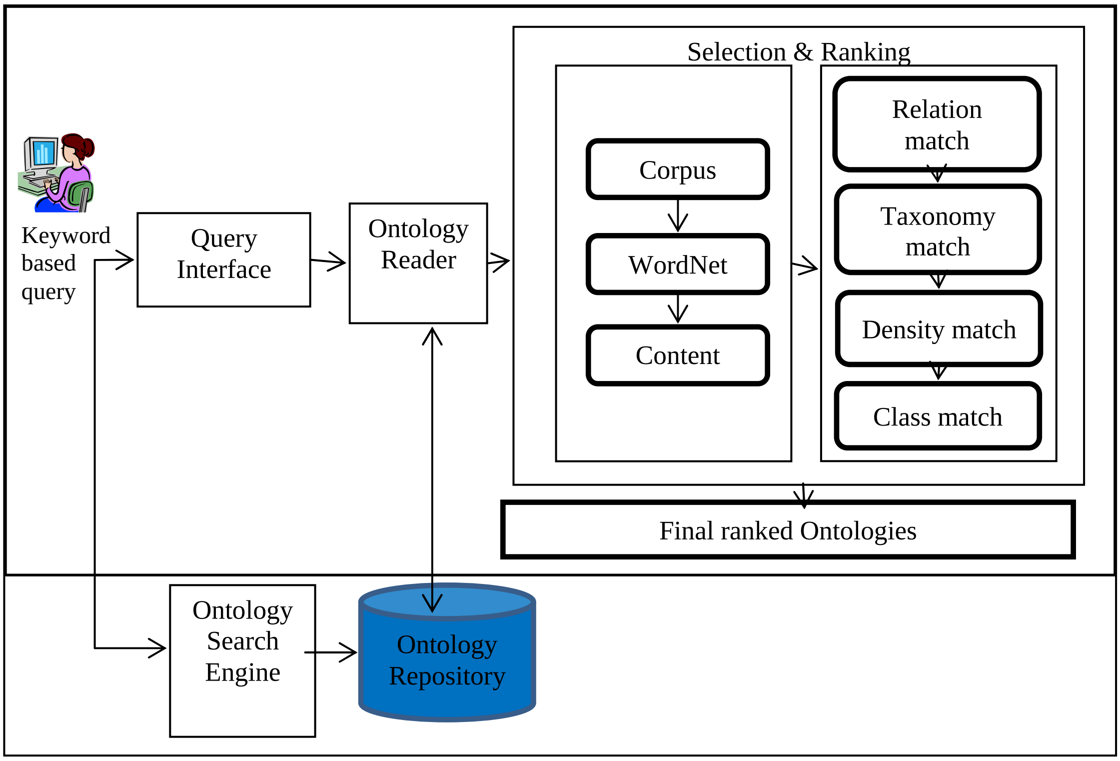

Fig 1: Architecture of the Proposed System

The ranking module reads the internal structure of the ontology and applies the content based ontology ranking in it. The obtained result will be given to the Content-OR module as the input and again ranked by using some new measures which includes Relation Match Measure, Taxonomy Measure, Class Match Measure and Density Match Measure (Reduced model). The output of the Content-OR rank is the final ranked list of ontology URIs. This ranked list of retrieved ontologies will be returned to the user or an agent.

B. Search Term Expansion

The search term expansion is done by using the WordNet. The keyword based query is accepted from the user as the input. The query contains the terms to search for. In Swoogle while searching for ontologies more terms need to be added in the Swoogle query string, other than those given by users. These extra query terms can be obtained from WordNet. The WordNet helps to expand the search term as its synonym; meronyms; hyponyms etc. The expanded search term is given as the input to the Content ranking.

C. Obtaining Ontology Score using CMS and LMS

The Class Match Score (CMS) for each ontology is obtained by matching the search terms with the class labels within them. The ontologies are also analyzed to see if any literal text matches the potential class labels [8].

Definition 1: *Let O be the set of ontologies to be ranked, C the set of potential class labels obtained from the corpus and n is the number of terms collected from the corpus.*

$$\text{CMS}\,[o \in O] = \sum_{i=1}^{n} I(C_i, o) \times 5\log(n+2-i) \qquad (1)$$

$$I(C_i, o) = \begin{cases} 1: \textit{if o contains a class with label matching } C_i \\ 0: \textit{if o contains a class with label which contains } C_i \\ 0.4: \textit{if } C_i \textit{ does not appear in any of o's class labels} \end{cases}$$

The literal text match score (LMS) is the same as Class Match Score (CMS), except that I (C_i, o)is now 1 if the ontology (*o*) contains text that matched a given term (C_i) and 0 otherwise. The total score for each ontology is a combination of these scores.

$$\text{Total} = \alpha \times \text{CMS} + \beta \times \text{LMS} \tag{2}$$

Where α &$\beta$are weights.

D. Ranking Ontologies using Reduced Model

The reduced model uses only four measures including Class Match Measure (CMM), Taxonomy Measure (Taxo), Relation Match Measure (RMM), and Density Measure (DEM). The user has the choice of selecting the measures.

1) *Class Match Measure (CMM):* The Class Match Measure (CMM) [2] is used to measure the coverage of ontology for a user supplied search term. It checks for classes in each ontology that having matching (exact match or partial match) labels to the search term.

Definition 2*: Let C[o] be the set of classes in ontology o & S is the set of search Terms.*

$$E\,[o, S] = \sum_{c\,\in C[o]} \sum_{s\,\in S} I(c, s) \tag{3}$$

$$I(c, s) = \begin{cases} 1 \; : \; if\ label\ (c) = s \\ 0 \; : \; if\ label\ (c) \neq s \end{cases}$$

$$P\,[o, S] = \sum_{c\,\in C[o]} \sum_{s\,\in S} J(c, s) \tag{4}$$

$$J(c, s) = \begin{cases} 1 \; : \; if\ label\ (c)\ contains\ s \\ 0 \; : \; if\ label\ (c)\ not\ contains\ s \end{cases}$$

Here, E [o, S] and P [o, S] are the number of classes of ontology o that have labels that match any of the search term s partially or exactly respectively. The Class Match Measure for ontology o with respect to search term s is,

$$\text{CMM}\,[o, S] = \propto E[o, S] + \beta P[o, S] \tag{5}$$

Where α and β are weight factors.

2) *Taxonomy Match Measure (Taxo):*The taxonomy matching is a type of relation matching that measures the semantic similarity of a taxonomy relation. It only considers the IS-A relation which is a hierarchical relation. Taxo [1] is the measure that evaluates the degree of semantic similarity of a taxonomy relation in a taxonomy structure.

Definition 3*: Let Rs, Rc be a binary relation where Rs =$\langle IS - A, sp, sc\rangle$, sp Є {$s_1$, synomys of s_1} and sc Є {s_2, synonyms of s_2}, s_1 and s_2 are search terms and s_p is the parent of s_c. Rc=$\langle IS - A, ci, cj\rangle$, where ci and cj are the concepts in the ontology o. ci is the parent of cj.*

$$\text{Taxo}\,[o, s_1, s_2] = \text{Sim}_{taxo}(o, R_s) = \text{Sim}_{taxo}(o, S) = \frac{F(o,S)^2}{\text{Distance}(o,S)} \tag{6}$$

$$\text{Sim}_{taxo}(o, S) = \sum_i \sum_j \text{Sim}_{taxo}(o, ci, cj)$$

$$\text{Sim}_{taxo}(o, S) = \sum_i \sum_j \frac{\text{Sim}_c(o,ci) \times \text{Sim}_c(o,cj)}{\text{Distance}(o,ci,cj)} \tag{7}$$

Where F (o, S) is the concept match and Distance (o, S) is the distance between concepts.

3) *Relation Match Measure (RMM):*The Relation Match Measure [1] (RMM)evaluates the degree of semantic similarity of a relation Rs between search terms in an ontology o. It considers concept match, neighbor match, relation label match and distance between concepts. It is also the inverse of semantic distance between two concepts.

Definition 4: *Let Rs, Rc be a binary relation as follows: Rs = <r, sd, sr > r: label of a relation Rs. sd Є {s_1, synonyms of s_1} and sr Є {s_2, synonyms of s_2}. s1, s2: search terms. Rc = <r, cd, cr> r: label of a relation Rc between concepts cd, cr.*

$$\text{RMM}\,[o, s_1, s_2, r] = \text{Sim}_r\,(o, S, r) = \sum_d \sum_r \text{Sim}_r(o, c_d, c_r, r)$$

$$\text{Sim}_r(o, c_d, c_r, r) = \frac{(F(o,c_d) - N(o,c_d))(F(o,c_r) - N(o,c_r))(1 + L(o,c_d,c_r,r))}{\text{Distance}(o,c_d,c_r)} \tag{8}$$

Where F (o, c) is the concept match, N (o, c) is the neighbor match, $\mathrm{Distance}(o, c_d, c_r)$ is the minimum path length between c_d and c_r and L (o, S, r) is the relation label match.

4) Density Measure (DEM): Density Measure [7] estimates the semantic richness of the concepts of interest. It considers the direct relations of subclasses, super classes and siblings for calculation.

Definition 5: *Let S = {S1, S2, S3, S4} = {Relations[c], super classes[c], subclasses[c], Siblings[c]}*

$$\mathrm{dem}(c) = \sum_{i=1}^{4} w_i\,|S_i| \tag{9}$$

$$\mathrm{DEM}\,(o) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{dem}(c) \tag{10}$$

Here, w_i is a weight factor set to a default value of 1 and k = E (o, S) +P (o, S) which is the number of matched classes in ontology o.

5) *Semantic Similarity Measure (SSM):* The semantic similarity measure [SSM] [7] checks the proximity of the concept of interest present in the ontology structure. It is calculated by counting the number of links that connects a pair of concepts.

Definition 6*: Let c_x, c_y are element of classes[o], and there is a path exist between c_x and c_y where p ε P.*

$$\mathrm{Ssm}\,(c_x, c_y) = \begin{cases} \dfrac{1}{\mathrm{length}(\min w \in W\{cx \to cy\})} & : \text{if } x \neq y \\ 1 & : \text{if } x = y \end{cases} \tag{11}$$

$$\mathrm{SSM}\,(o) = \frac{1}{K}\sum_{x=1}^{n-1}\sum_{y=x+1}^{n} \mathrm{ssm}(cx,\ cy) \tag{12}$$

Where n = number of matched classes, and $K=\sum_{k=1}^{n-1} k$

Total score of reduced model is,

$$\mathrm{Total}[o] = \alpha\,\mathrm{CMM}\,[o, S] + \gamma\,\mathrm{SSM}\,[o, S] + \delta\,\mathrm{RMM}\,[o, s_1, s_2, r] + \varepsilon\mathrm{DEM}\,[o, S] \tag{13}$$

Where $\alpha + \gamma + \delta + \varepsilon = 1$.

IV. EXPERIMENTS AND RESULTS

In this section, the experiments were performed by using reduced model. The ontologies are downloaded for the search terms “book” from the semantic web. The ontologies are downloaded using the Swoogle semantic search engine and is stored in the local ontology repository. Table I shows the list of candidate ontologies downloaded from Swoogle.

TABLE I
List of Candidate Ontologies

Ontology	**Ontology Title**
Book0	http://www.hackcraft.net/bookrdf/vocab/0_1/
Book1	http://swrc.ontoware.org/ontology
Book2	http://morpheus.cs.umbc.edu/aks1/ontosem.owl
Book3	http://swrc.ontoware.org/ontology/portal
Book4	http://www.aktors.org/ontology/portal
Book5	http://ebiquity.umbc.edu/ontology/publication.owl
Book6	http://lsdis.cs.uga.edu/projects/semdis/opus
Book7	http://purl.oclc.org/NET/nknouf/ns/bibtex

A. *Content Ranking Score*

The search query is accepted from the user and is expanded by using the WordNet software. Fig. 3 shows the WordNet expansion for the query “student”. For the search terms, retrieve a corpus from the web that covers this domain and using the expanded search terms; get a list of potentially relevant ontologies from the Swoogle. Fig.2 shows a small portion of an ontology.

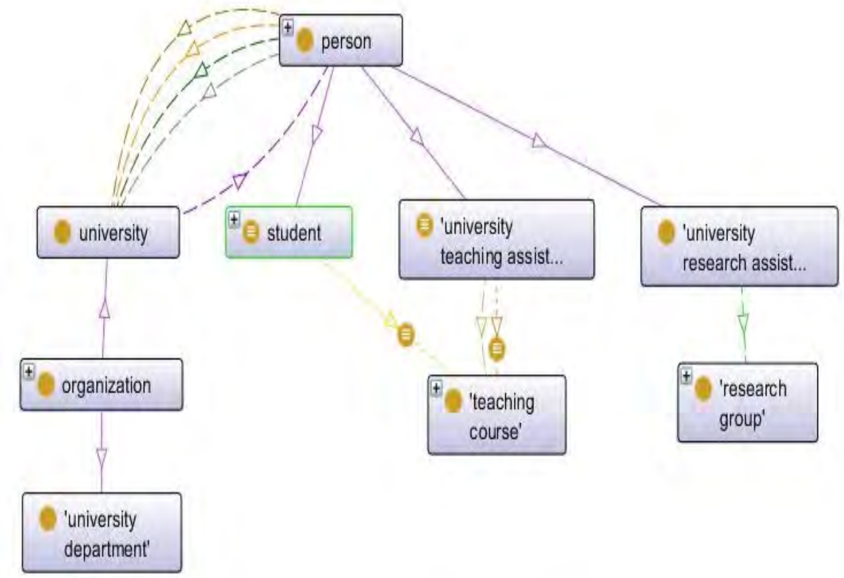

Fig 2: A small Portion of Ontology Graph from Protege

The corpus will then be analyzed to identify domain-related terms to use for evaluating the existing ontologies in terms of how well they cover the domain of interest. Using a representative corpus allows terms to be extracted using term frequency measures (Tf-idf). The terms which get the highest Tf-idf score from this corpus can then be considered as potential concept labels. This system uses the top 50 words of such an analysis. An ontology which has more class labels that match these words is deemed more suitable by the system and is therefore ranked higher than others. Each ontology is then ranked according to how many of these new terms match class labels within them; the class match score (CMS) and the Literal Match Score (LMS). The total value is obtained by adding these two scores including the weight factors α and β and the ontology with highest value will be ranked first.

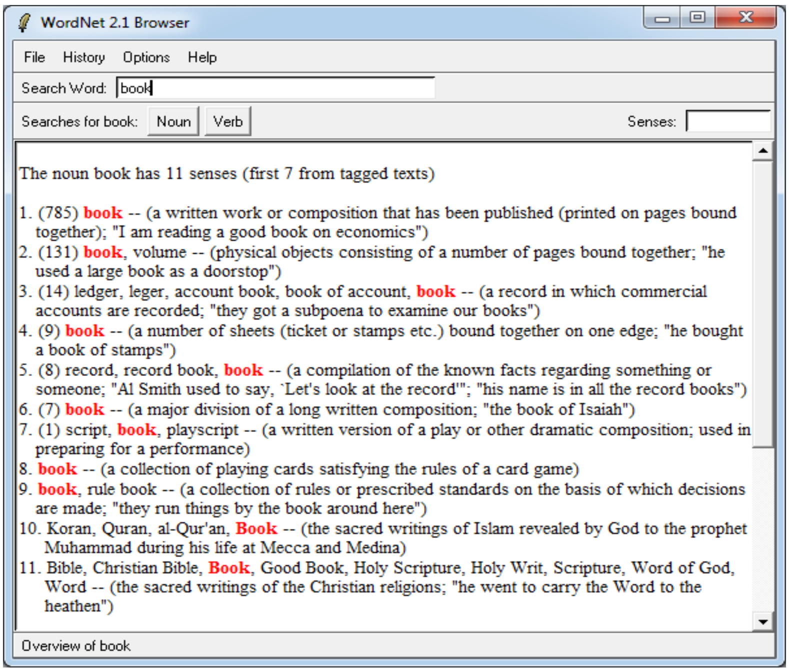

Fig 3: WordNet Expansion for the term "book"

Calculation of CMS & LMS for the ontology "book0"

Let the value of α = 0.8 and β = 0.2. So the total value will be,

Total = α × CMS + β × LMS

= 0.8 (0.00693) + 0.2 (0.0693)

= 0.019408

TABLE II
Ontologies Ranked Using Content Rank

Id	File Name	CMS	LMS	Total
1	Book2.owl	36.94074	0.450551778	29.6427021
2	Book1.owl	0.6708668	0.273823172	0.5914581
3	Book4.owl	0.587061465	0.268064618	0.523262143
4	Book3.owl	0.49748528	0.231248647	0.246048555
5	Book6.owl	0.13251996	0.203876868	0.146791354
6	Book7.owl	0.03468446	0.144518584	0.0566512868
7	Book5.owl	0.01455609	0.103972077	0.0324392878
8	Book0.owl	0.006693147164	0.06931472	0.0194081217

B. *Content-OR Ranking*

The ontologies which are ranked by using the Content Based Ontology Ranking model are again ranked by using the Ontology Rank which considers the reduced model proposed by Jinsoo Park et.al [1]. It mainly includes 4 measures which are Class Match Measure (CMM), Density Match Measure (DEM), Relation Match Measure (RMM), Taxonomy Measure (Taxo) .The measures can be changed according to the application.

According to the equations 5, 8, 10 and 12, the values are obtained for the ontology and are applied in the equation (13). Here the weight factors α, δ, γ, ε are given three sets of values which are based on AKTive rank weight setting, and Equivalent weight setting and Relation focused weight setting. Now the total value is find out by the equation,

$$\text{Total} = \alpha \text{ CMM } [o, S] + \delta \text{ SSM } [o, S] + \gamma \text{ RMM } [o, s_1, s_2, r] + \varepsilon \text{ DEM } [o, S] \quad (14)$$

These value is obtained for all the ontologies and according to the Total value obtained for each ontology they are ranked. Table III. shows the ranked list of ontologies based on three weight settings.

TABLE III
Ranking Based on the Three Weight Settings

AKTIVE RANK		EQUIVALENT WEIGHT RANKING		RELATION FOCUSED RANKING	
File Name	Total Similarity	File Name	Total Similarity1	File Name	Total Similarity2
Book2.owl	0.528842568	Book6.owl	0.375714272	Book2.owl	0.111678258
Book6.owl	0.431150734	Book2.owl	0.36	Book1.owl	0.102402538
Book1.owl	0.39932698	Book1.owl	0.27466667	Book3.owl	0.09931286
Book3.owl	0.3974143	Book3.owl	0.213	Book6.owl	0.08958686
Book7.owl	0.09270409	Book5.owl	0.0595833324	Book5.owl	0.07230729
Book5.owl	0.09140624	Book7.owl	0.0557142869	Book7.owl	0.0714540854
Book0.owl	0.0395679027	Book0.owl	0.0466666669	Book0.owl	0.04865432
Book4.owl	0.03928889	Book4.owl	0.0284444448	Book4.owl	0.0391911119

Since the ontologies which are ranked once are again ranked by using the Content-OR model, the ranking will be more accurate. Also the result of the ranking shows accuracy.Figure 4. shows the Performance graph for the three weight setting which are AKTive rank, Equivalent weight setting and Relation focused weight setting.

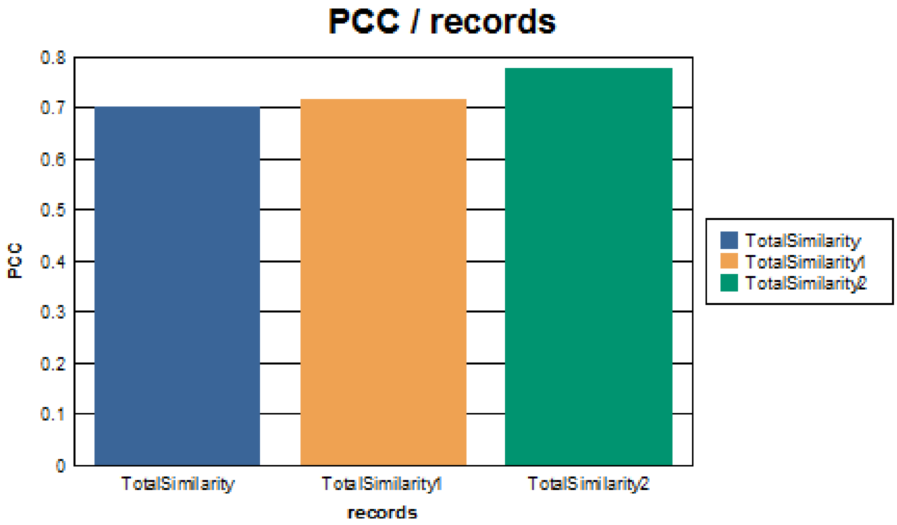

Fig 4: Performance Graph for three weight settings

The graph shows the performance of the ranking obtained based on the pearson correlation coefficient. It shows that the total similarity 2 in which the relation focused weight setting is applied shows the best ranking .

V. CONCLUSION

In this paper a combined ranking algorithm integrating two other ranking algorithms is proposed. It uses the semantic similarity mechanism rather than lexical level matching in order to rank ontologies. It helps increasing the accuracy of the ranking of ontologies in semantic web. Since the model ranks the ontology using content ranking and the reduced model, it assures good results on the ranking of the ontologies. The performance graph shows a better value when the relation focused weight setting is used for ranking. Although it combines the advantages of two ranking algorithms and the initial results looks good, more experiments have to be carried out before making any conclusive remarks about the Content-OR Ranking.

REFERENCES

[1] Jinsoo Park, Sunjoo Oh, Joongho Ahn, Ontology Selection Ranking Model for Knowledge Reuse, Expert Systems with Applications 38, 5133–5144 (2011)

[2] Alani, H., & Brewster, C., Ontology ranking based on the analysis of concept Structures, In Proceedings of the third international conference on knowledge capture (K-CAP'05), Banff, Canada 2005.

[3] Buitelaar P., Eigner T., & Declerck T., OntoSelect: A dynamic ontology library with support for ontology selection, In Proceedings of the demo session at the international Semantic Web conference, Hiroshima, Japan. 2004

[4] Vandana Dhingra, Comparative Analysis of Ontology Ranking Algorithms, International Journal of Information Technology and Web Engineering, 7(3), 55-66, July -September 2012

[5] Ding. L., Finin. T., Joshi. A., Peng. Y., Cost R. S., & Sachs J, et al. Swoogle: A Search and metadata engine for Semantic Web, In Proceedings of the thirteenth ACM conference on information and knowledge management, 2004

[6] Ding L, et al. Finding and Ranking Knowledge on the Semantic Web, Proceedings of the 4th International Semantic Web Conference 156-170107

[7] Harith Alani, Christopher Brewster and Nigel Shadbolt, Ranking Ontologies with AKTive Rank,2007

[8] Matthew Jones and Harith Alani, Content-based Ontology Ranking, 9th International Protégé Conference, July 2006

[9] Samir Tartir and I. Budak Arpinar, Ontology Evaluation and Ranking using OntoQA,2007

[10] Zhiguo Ding and Zhengjie Duan, Improved Ontology Ranking Algorithm Based on Semantic Web, 978-1-4244-6709-9/10/$26.00 ©2010 IEEE

[11] R. Subhashini, J. Akilandeswari, V. Sindhuja, A Review on ontology Ranking Algorithms, International Journal of Computer Applications (0975-8887), November 2011

[12] Harith Alani. Position paper: Ontology construction from online ontologies, WWW2006, Edinburgh, UK, May 22–26, 2006

[13] Janez Brank, Marko Grobelnik, Dunja Mladenic, A survey on ontology evaluation techniques.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, Technical report, Stanford University, 1999

[15] Patel C, Supekar K, Lee Y, and Park E. Ontokhoj: A semantic web portal for ontology searching, ranking, and classification,. In Proc.5th ACM Int. Workshop on Web Information and Data Management, pages 58–61, New Orleans, Louisiana, USA, 2003.