Constructing Virtual Documents for Keyword Based Concept Search in Web Ontology

Sapna Paliwal[1],Priya.M[2]
[1]School of Information Technology and Engineering,
VIT University,
Vellore-632014,TamilNadu, India
pmmt77sapnaa@gmail.com
[2]School of Information Technology and Engineering,
VIT University,
Vellore-632014,TamilNadu, India
sumipriya@gmail.com

***Abstract*-Web ontologies are structural frameworks for organizing information in semantics web and provide shared concepts. Ontology formally represents knowledge or information about particular entity as a set of concepts within a particular domain on semantic web. Web ontology helps to describe concepts within domain and also help us to enables semantic interoperability between two different applications by using Falcons concept search. We can facilitate concept searching and ontologies reusing. Constructing virtual documents is a keyword based search in ontology. The proposed method helps us to find how search engine help user to find out ontologies in less time so we can satisfy their needs. It include some supportive technologies with new technique is to constructing virtual documents of concepts for keyword based search and based on population scheme we rank the concept and ontologies, a way to generate structured snippets according to query. In this concept we can report the user feedback and usability evolution.**

Keyword- Ontology search, Snippet generation, Virtual document, Indexing, Ontology ranking

I. INTRODUCTION

Web ontology is widely used term in information science. Ontology is generally used for information related to particular object as a set of concepts and present within domain. All these information organized in semantic web then we can say that as web ontology. Thesis concepts are related to each other in ontology and help to search related concepts in less time. We can use ontology to model a domain and to share the concepts. Domain entity described by these shared concepts. It also help user to provide interpretability between two different applications on particular semantic that is called semantic interpretability. Ontology is a specification of shared conceptualization because it shared object and concepts as well as their properties and relation. We generally use this concept to reuse ontologies. In computer science web ontology describe as set of types, properties and relations on web. To integrate data on any web application we use semantic web. Resource Description Framework (RDF) is design as metadata model. It is used to for matte data that is present on the semantic web. All this information is describe by triple graph. It is generally used to describe the concepts and model the information that is implemented in the web resources using some syntax notation and serialization format. The RDF data model is similar to entity relationship in the form of subject predicate and object expression. These expressions are known as triples in RDF terminologies. The World Wide Web consortium has developed guideline for publishing RDF vocabularies on web. It maintains it as tool for maintaining a vocabulary of term. For vocabulary management task they introduce new language that is called RDF Vocabulary Description Language (RDFS).

Resource description framework schema (RDFS) helps us to describe web ontologies and also Web Ontology Language provides us shared concepts. RDFS is most commonly used schema language on semantic web technology stack. It is easy to use so many RDF vocabularies describe in basic RDFS. It is similar to the universal translator. It allows data created by different teams for different uses at different time to be connected using RDFS. It will put some restriction how to use vocabulary and some triples are help to configure these restrictions. For describing domain entities we use classes and properties. It also help to provide semantic interoperability between two different applications. Semantic interoperability means ability to transmit data unambiguous shared meaning between two computer systems. It also helps to improve machine computable logics, inference, and knowledge discovery in information system. Semantic interoperability is simulation transmission of meaning with the data. It added metadata and liked each data element to controlled, shared vocabulary. Data is transmitted in to information packages that are independent of any information system. All

this above process depends on reuse of ontology that are existing or to extend it at the time of developing new applications.

Ontologies define context and also provide us vocabulary that is help to describe our web pages. Mostly application developer's basic steps are ontology search. Semantic search improve the accuracy of search by understanding user intent and contextual meaning of the term. Semantic search system use various ways to improve their search result like context of search, location, intent, synonyms, specialize queries etc. mainly two form of search is their navigational and research. In navigational search, user use search engine for navigation as navigational tools. In another research search user is provide semantic search a phrase as an object to get some research information that is necessary for user. This search engines not return particular document for user to satisfy their need beside that it returned set of documents related to their query so user can able to find their result what they want. Search engines are programmed to search according to the user keyword query and return the list of documents for that particular keyword. Every search engine use proprietary algorithms to response user according their desires. Search engines use various automated software programs that is called as spiders to survey the web or their database. Search engine reply to user matched concept according to user queries. Generally the result of the search is metadata and RDF description that is related to the user keyword query. Output of this type of search cannot satisfy user needs. To overcome these problems and provide user better result we develop a novel keyword based ontology search engine. It retrieves the concepts that almost user want and the concept return is matched with the textual description given by the user. It also provides all the related ontologies so the user can easily compare and find the result what they want. It helps to effectively fulfill the user needs. Output is given to the user is base on popularity of the entity and according to query relevance. According to the popularity of particular concept we can rank the output of the search engine. Popularity we can find by the most visited web search or collect large number of data from the semantic web. It help to user can easily find their result what they are want. It is a database system that is design to index and categorized address. Search engine spiders crawl from internet and identify the web page. Once search engine spider identify the web site then it index it is contents. Provide URL available to the users so user can easily search it. When user search any query from the search engine that means he is asking to scan their database and retrieve the relevant result to the user. The result of the system is associated with the structured snippets relevant to the query. It give all information to the user the concept is matched or not with the keyword query entered by user or not and also provide meaning full information to the user. In the other search engines their mass data is present so more probability to find irrelevant information. Search engine result varies according to the search engine to search engine like size, speed, content. It is also varies according to the ranking criteria used by the search engine. To rank each concept search engine follow different criteria such as they generate most popular web pages on the top of their list.

The term search engine describes both crawler based search engine and human powered directories. In crawler based search engine search done by the crawler or the spider on the web. It returns the web pages or documents that user want. Another one is human based search such as open directory depends on the listing it give result. Most of the search engines having three parts. First part is spider that identifies particular page through visiting all pages in site and read it for more better result navigate to the user to the another page in same web site. We can also call it crawler for search engine. The result search by the spider it passes to the next part of the system that is index. It is like a large data that store copy of each web page and also call catalog. If the information stored in any web page is change then it is copy also updated according to the required changes. Search engine software is the main and third part of the search engine. It extract the records from the millions of information from the index according to the matches then rank all the information depends on some different ranking algorithms to provide user better satisfaction. It provides the restriction over the ontologies so user can restrict the search for better result. For ontology restriction some query relevant popular ontologies recommends to the user. Search engine rank the ontologies the most relevant one comes first then other that laser relevant to it. One of the main things in the ranking algorithms is location and the frequency of particular keyword on that web page. It is also called location frequency method. It provides facility to the user to compare ontologies to get better performance. The result return by the search engine is checked through their context and query relevant concepts. According to the output of the system it will return some RDF description that gives all information about the concept return. To facilitate the user it provide description of ontologies for better understanding. In the search engines main thing is to improve it is performance provide user effective result in the less time. Different search engine use different technique to improve their performance. In the search engine there is one program called spider it identify data from large storage it take more time compare to find data in particular popular places. In our system we provide restricted ontology search to reduce the search time. Early search engine make search few hundred of thousand documents and inquired only one or two thousands. Now a day's search engines index hundreds of millions of web pages and response tens of millions pages per day. If we search on particular area or ontology then it will be easy to search and improve system performance.

II. SYSTEM ARCHITECTURE

In the proposed system architecture user search information through the user interface, First client enter his keyword query on the web page. Once user enter query this request is pass to servlet that is design to process the request. Processing unit pass it to Data Access object request. Data Access object perform required operation on it then send it to database. According to user keyword query database retrieve the resultant information to servlet program. Servlet http response to the main part of the engine depends on the user demand it return object or relation to the user that help to efficiently satisfy user needs. To improve the performance we perform two search operation object search and concept search. Object search is goal to find the location of the object in the large data collection. Object search is done by the help of some classes with the help of index. In object search we search data in the form of object and stored this entire object in to the index. We also collect all the classes for particular object is return by the search. The ranking is done for all this classes according to their popularity of their objects in the indexes. The classes they are at top of the search are selected. For the user recommendation all the classes are grouped. In the concept search the result is provided by the search engine is conceptually similar to the keyword query entered by the user. In the concept search we index all the concepts return by the search result like ontologies and it is classes and properties. For better performance we use the inverted index concept and according to the user ontology restriction concept is return in the form of matched concepts to the user.

In user interface we create a user page using GUI, which will be the media to connect user with the through which client can able to give request to the server can send the response to the client, through this module we can establish the communication between client and server. Before client creation we check the user credential her by login page, we receive the username and password by the user and we will check in the database is that user have the credential or not to give request to server. Here also we can add new user through user registration by taking all the important details like user name, gender, username, password, address, email id, phone no from the user. Then we design the page for getting the user query then we will write the code in java to process and store the query request to the semantic storage. For object implementation we get the query request from the user and search result for that query from the semantic storage and send the result to the user in the form of some object may be the accurate expected result for the query writer, this way of searching in the semantic web is called object search, through which we can reduce the search timing of the user can able to give the useful response to the user. We design the database as the semantic web storage which could be given an actual result for the concept search. We also have to design the tables here as per the user query requirement and put the correct relation among those all tables and fields.

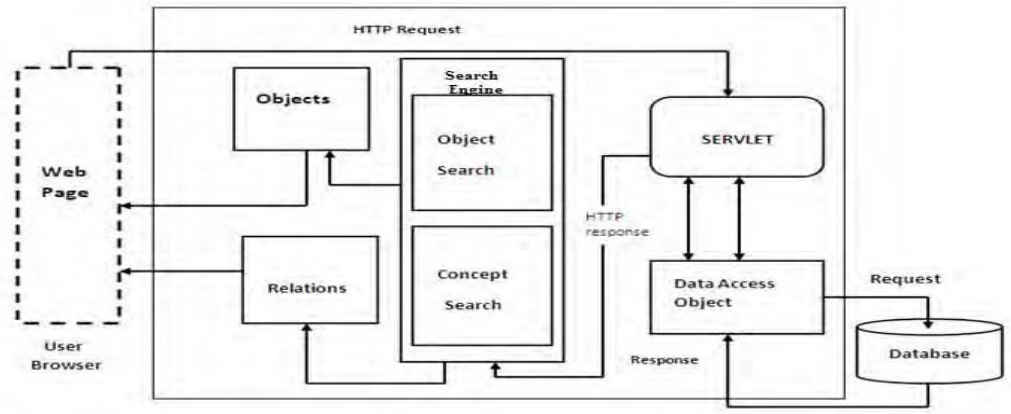

Fig. 1. Architecture of System

According to query request from the user and search for that query from the semantic storage by the finding relation between the words of the query. Here we will receive user query in the form of distinct words as per that all word we will search what may be the actual and possible relation among those word, after got the relation now we can search as per that relation in the semantic web and send the all possible relation result to the user. This system architecture based on Falcons concept search engine in this architecture multithreading crawler dereferences URI with content negotiation and download RDF documents, these documents parse by system.

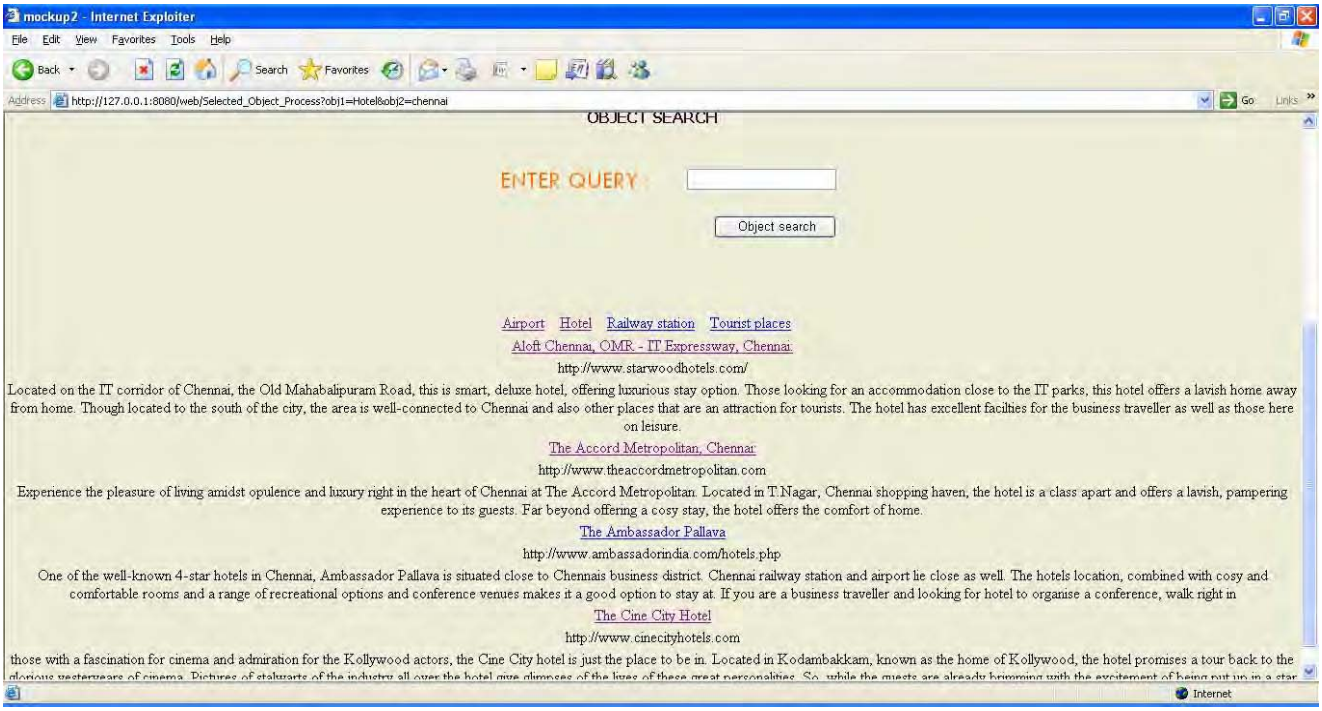

Fig 2. First result page for keyword query of object search.

All information retrieve from the semantic web like RDF triples and the document URI is store in quadruple store .we implemented in our project as MySQL database. For the computation of the data we design metadata analysis. To update the combine inverted index we used indexer. It helps to serve user interface. User can easily restrict ontology. It includes two inverted indexes. First one based on our storage information, we will get all information related to our concepts and ontologies. An inverted index is containing all information about ontologies and related concepts. By the help of this technique we can restrict the search or filter the result according to our needs. To provide the better performance to the user construction of virtual document included. Virtual document constructed for each and every concepts. Each virtual document contains the keywords that are fetching from the RDF description. In the inverted index contained the list of references to the document in to the large storage. It is a index data structure that is storage of references to particular documents. It help to retrieve the information for particular keyword search. Inverted index return two results. These two result combine together and find it is intersection for better result that is exactly concept return by the system. After we intersect the result according to popularity we rank the result. A popularity score is computed at the time of indexing. This information attached with particular concept. According to the user demands structural snippets is generated for each resultant concept that is stored in quadruple store. Recommendation of ontology is done by system according to top ranking concepts. According to the rank of concepts and also the similarity of context between the keyword queries are combined to rank concepts and virtual documents of concepts at searching time. According to user request of particular concept RDF description is loaded by browsing concepts component from quadruple storage and based on user demand presentation is done. Similar to user ontology request list of classes and properties that contains metadata are loaded by ontologies component loads and presented to the requested user.

III. VIRTUAL DOCUMENTS

The most important part of any search engine is gaining the understanding of user requirement in order to provide more satisfied result. Virtual documents are the dynamically generated documents that describe what explanation user want. All the web ontologies documentation use RDF and OWL language to build it is structure. It is help to combine the web ontology to the semantic web. In this document so many ontologies are stored and some time it is overlapped in the same domain. Virtual document are hypermedia document that generated according to the user demand. Ontology matching is very important concept for the user to provide interoperability between two different web applications. They are embedded in context of other documents. Virtual document are uses the term that are extracted by the concepts. Constructing virtual documents for particular model it is reflect artifacts of being describe. It is incorporate the result of analysis and discussions. All the terms that include in the virtual documents are available in web pages. It helps to provide keyword based search. By constructing virtual document we can save out space and it is also cost effective approach for the user. Using this user can easily match the ontologies that is also called linguistic matching. Virtual documents are document that can contain other documents for better description. Virtual document contains a set of

weighted words. It is also containing the relative information to the particular word with their local description to provide user more relevant information. Virtual document also contain type and subtype of the objects. Contain of the virtual document may be the different type of object.

Resource description framework is a framework to represent information on the web. It is also include design goal key concepts and data type etc. We needs the description formulation to describe the concepts for that we use resource description framework graph. It is also help user to perform neighboring operation for relevant information to get better performance and provide user satisfaction. In this system we can easily find the matched concepts and context based on the similarity of the documents. It is calculated by the vector space method then this information is used by the matching algorithms. Each virtual document having own concept and all this concept is describe by the RDF triples. This RDF triples help to describe it and give us terms based on this information we can construct virtual document on the semantic web. RDF graph is set of RDF triples. For construction of virtual document we need terms to get this terms we use local name and related values such as it is label etc. all this information like comment present in to the traditionally used search engine we takes this from there. Resource description framework is not easy to understand and describable in order to understand it is concept we use some extra information besides literal values. All this information helps us to find the relation between our word and virtual documents.

In this approach we show above, All type of information such as structural or knowledge of any particular entity are stored in the ontologies as a form of elements. It is having own references to mapped it from its URI references. By the help of these references we can easily fetch it when it is required. The information related to the URI references is present into the ontology web language ontologies. Ontologies contain all information related to particular concept such as it is name, type. All this information is used for fragmentation of the identifiers. Structural information contains URI references and it is relations to the other information such as other labels and comments. Similarity between URI references and keyword are help to construct virtual documents so the user can easily detect their result. Similarity between keyword and URI references decide construction of virtual documents. All these constructions depend on the approaches used if we use similarity based approach then we have to calculate similarity between the information. After calculate it use this idea into our approach. One of the widely use approach is matching approach that is based on the structural similarity also called key feature of this approach. All these information are basis of the ontology matching. Ontology matching is the similar to the scheme matching approach. To match the ontologies the ontologies matching approach is very cheap compare to other techniques used. To improve the system performance we need to know about characteristics of the ontologies these characteristics help us to create or construct virtual documents for scheme matching. Virtual documents are documents that generate dynamically according to user requirement to satisfy their needs. It is mainly help user to generate dynamic web pages according to user demands to fulfill their requirements. For reuse the information present into the existing system we also use this dynamic creation of the documentation. These documents contain all the necessary information and also related references to provide more information about the concept. Virtual document provide user fast access to the search and help them to get more descriptive information about any object. User can easily restrict the ontologies to support system technique to fast information retrieval. This information help user to find ontology is match with other or not that is called ontology matching. To construct this documents we needs RDF graph structure.

IV. CONCEPT RANKING

Concept ranking of particular concept is calculated by its popularity or query relevance. Ranking of concept is very usable idea that efficiently satisfies user needs and gives sufficient information about particular object. Different way to find the rank of concept one is popularity. It is the existing approaches to study ontology structure evaluation of concept with structural measures. For that some algorithms are wildly used such as page rank like algorithms or graph centrality. Different search engines use different algorithms for concept ranking according to their requirements. Some algorithms performance is really very good but some are not perform well when we use on particular ontology. Performance of algorithms is differing based on the architecture of the search engine. For the new developer they want to use the concept of interoperability between the two different applications. Based on the popularity calculated by the system and the data set collected from the large database of the web we can rank the concepts. According to the value calculated by system we can give higher rank to the concept that is more popular. RDF document is created for each particular concept in the system. This document also contains classes. These classes help to insatiate the other concepts inside the RDF documents. This documents help to predict their type and objects of particular classes. Mainly four feature of the concepts and relations is importance of the concepts is increased when their more relation are present. If there are more concepts are presents, if higher relation weight to other concepts, if weight is higher and start with the more important concepts.

Each concept that is return by the system index having own RDF documents. Inside these documents we can easily institute the classes related to that concepts for better description. Each document having various RDF triples. This RDF triples are different for each classes. All the classes has it is own objects to describe it

properly. The notation of the classes and the documentation is c and d. if at the place of classes we use properties then we can predicate it by c. it is provide vocabulary to describe the classes better way. Each container of classes having it is type, member and so we can manage it properly. Some container manages all information in order and some are unordered. For manage more information we took data from the semantic web. According to all these information we decide which one is more popular or not for user point of view to satisfy their needs. There is no specific condition for the vocabulary into the container but all this vocabulary contains only necessary and basic information. For the rank any concept we mostly use here only two techniques one is query relevance and another one is popularity of the concepts.

V. STRUCTURAL SNIPPETS

In this part of the paper we are going to explain how to generate structural snippets for the output return from the system according to the query entered by the user demands. Structural snippets generate in order to provide user better result so they easily understand it. Generating structural snippets is the final step of the system for presenting the result to the user. It helps to the user to clarify how the concept is match with the keyword that user entered and generating snippets are generated according to the user query only. All this snippets that are generated by the system it help user to understand the result and also helps them to find relevant information in the less time. It provides result in less time and easy to find information related to the any ontologies or any specific ontology for better performance. For the generation of snippets there are so many methods but we use threading method in this method we use the threading concepts. In this concept thread is the fundamental unit for this process. After process it we can rank these threads according to the some criteria. This process helps us to find the more popular snippets. We use in our system two methods for generating snippets one is to rank the threads according to that generate threads and another one is PD thread that is a fundamental unit of the generating a snippets.

In the first method structured snippets is generated by the system according to the user demands and depends in it is query keyword to provide better description. Each concept has three threads then we have to rank all this according to the user entered keyword query. After Appling ranking algorithms we can easily rank the threads. In this algorithm first we find the rank of each and every thread that is related to that concept. After finding this rank of every thread separates the high rank threads. If the thread that is related to that concept is not in the high rank then does this process again for more time till it will not present. But in the second method thread is the main and basic unit for the generating structured snippets. When system generate the result in the form of graph but it is too large to understand properly to the user and cannot able to find what the relation to all these and the main concept. To overcome these problems we separate some part of the graph that is necessary for the user to understand the concept. For that the RDF triple is not suitable for the description of the concepts. Blank node is use to connect the remaining nodes. It is not give any important information about the entities.

VI. EXPERIMENTS

In our experiment we show the work in the form of usability of the system and also the feedback of the user to our system. We collect this information after complete our work from the users and store also the feedback of the system. For the usability of the system except our system other search engines like Swoogle is very widely used and famous search engine for ontology search. These search engines also evaluated by the references. We are having some expertise of this area and found the result whatever ontologies they want to find they do it by themselves. Each ontology they select is contain it is own classes and the properties to describe the concept. According to their experience they help us to prepare system usability scale. These all information in that SUS score show that result of our system is more than 70 but the Swoogle is around 45. The result shows that our search engine gives better information and satisfaction to the user. These usability scales contain statement like uses, complexity of the system, frequently uses, user confidence, quick response, inconsistency, integrity of the function in the system. According to the result our search engine performs consistently better than Swoogle. Result is found based on the questionnaires prepared.

User feedback collected from the user they use it is as a experiments. Users are very impressed with the some feature like user interface, ontologies restriction and ontologies matching. it is help to provide user information very fast than other systems. It gives more detail in to the result to better understanding of the user. But some user suggests us to include more ontologoies in the recommendation for better performance. User feel comfort with the feature that generate the query relevant structured snippets that is help user a lot to understand the concepts and it is relation with the other objects. It is better to describe in the form of the graph. With this feedback we get some interesting opinions like clustering all ontologies into one is save user time and it should be design according to the normal user rather than researchers. We also did some related works such as a comparison of some widely used ontology search engines. It is shown below.

TABLE I

Comparision of ontology search enginesSearch engine	Input query	Keyword refinement	Basic information	Metadata	Query relevance
New search engine	Lucene syntax	Ontology selection	URI, label, type	Document frequency	Generating snippets
Watson	yes	HTML from	URI, label, comment	Size, language	Matched concept
Swoogle	Lucene syntax	Query syntax	URI, literals, type	rich	Matched literals, matched concepts
SQORE	yes	no	URI	no	Ranking score

Table 1 shows that each search engine having different way to organize their function in order to get better result. We compare here these search engines based on their query type and result presentation. In query type we include keyword that is type by the user and refinement of these queries. In result presentation of the concept includes the basic information of the output and also include metadata information of it is concepts. It is also contain query relevance that means according to query it generate structural snippets. For better compare these search engines we include ranking algorithms also to find better result in less time. Different search engine use their various ranking algorithms such as SQORE uses WordNet based algorithms to rank the concepts, Swoogle applies PageRank like algorithms and proposed search system use query relevance and popularity. In the Swoogle and Watson they already ask to the user about the search what they want classes or properties. But in new proposed system we enable user to restrict the search by restricting the ontologies. By the comparison to the other search engine it is clear that new proposed systems performs better and give more user satisfaction in their result.

VII.CONCLUSION

In this paper, we have introduced new way to search object based and concept based ontologies using Falcons concept search and have detailed design and its implementation. In this system we constructed virtual documents for keyword based object search and integrate concept level search and ontology level search by recommending ontologies. This system facilitates us to filter the ontologies according to user needs. It helps to user easily interact with the system and quickly find desired concepts, object and ontologies. It includes several supportive techniques like inverted index structure, a method of constructing virtual documents of concepts. This concepts includes the names of associated properties and related entities, a method of generating query relevant structure snippets and a way to rank concepts, objects and ontologies based on their popularity on the semantic web as well as their relevance to keyword queries. In the system for each concept returned, its type, label and a query relevance structured snippets are provided to help users quickly determine its relevance to their needs. The concepts returned and their structured snippets, user can quickly learn the relevance and characteristic of ontologies and user can easily compare ontologies. According to the user demand RDF description and ontologies also provided. In the system we can compare various aspects f ontology search engine and performed a usability evaluation. in this proposed system user can quickly compare ontology and determines whether these ontology satisfy their needs by checking query relevant concepts as well as their context. In the future work, we are interesting to consider other metrics for ontology evaluation and recommendation and we can investigate different query types besides keywords.

REFERENCES

[1] Castano, S., Antonellis, V.D., De Capitani di Vimercati, S. Global Viewing of Heterogeneous Data Sources. IEEE Trans. Knowl. Data Eng. 13(2) (2001), 277-297.
[2] M. d'Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, and E. Motta, "Watson: Supporting next generation semantic web applica- tions," in Proc. IADIS Int. Conf. WWW/Internet, 2007, pp. 363–371.
[3] L. Ding, T. Finin, A. Joshi, Y. Peng, R. Pan, and P. Kolari, "Search on the semantic web," IEEE Comput., vol. 38, no. 10, pp. 62–69, Oct. 2005.
[4] C. Anutariya, R. Ungrangsi, and V. Wuwongse, "SQORE: A framework for semantic query based ontology retrieval," in Proc. 12th Int. Conf. Database Syst. Adv. Appl., 2007, pp. 924–929.
[5] Y. Qu, W. Hu, and G. Cheng, "Constructing virtual documents for ontology matching," in Proc. 15th Int. World Wide Web Conf., 2006, pp. 23–31.
[6] C. Watters, "Information retrieval and the virtual document," J. Amer. Soc. Inf. Sci., vol. 50, no. 11, pp. 1028–1029, Aug. 1999.
[7] X. Zhang, H. Li, and Y. Qu, "Finding important vocabulary within ontol- ogy," in Proc. 1st Asian Semant. Web Conf., 2006, pp. 106–112.
[8] X. Zhang, H. Li, and Y. Qu, "Finding important vocabulary within ontol- ogy," in Proc. 1st Asian Semant. Web Conf., 2006, pp. 106–112.
[9] Do, H.H., and Rahm, E. COMA - A System for Flexible Combination of Schema Matching Approaches. VLDB 2002, 610-621.
[10] Salton, G. and McGill, M.H. Introduction to Modern Information Retrieval. McGraw-Hill 1983.

[11] Stoilos, G., Stamou, G., and Kollias, S. A String Metric for Ontology Alignment. International Semantic Web Conference 2005, 623-637.
[12] Sure, Y., Corcho, O., Euzenat, J., and Hughes, T. (eds.). Proceedings of the 3rd International Workshop on Evaluation of Ontology Based Tools (EON 2004). CEUR-WS Publication 2004.
[13] van Hage, W.R., Katrenko, S., and Schreiber, G. A Method to Combine Linguistic Ontology-Mapping Techniques. International Semantic Web Conference 2005, 732-744.
[14] Watters, C. Information Retrieval and the Virtual Document. JASIS 50(11) (1999), 1028-1029.
[15] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, "Sindice.com: A document-oriented lookup index for open linked data," Int. J. Metadata Semant. Ontologies, vol. 3, no. 1, pp. 37–52, Jan. 2008.
[16] C. Patel, K. Supekar, Y. Lee, and E. K. Park, "OntoKhoj: A semantic web portal for ontology searching, ranking and classification," in Proc. 5th Int. Workshop Web Inf. Data Manage., 2003, pp. 58–61.
[17] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, "Sindice.com: A document-oriented lookup index for open linked data," Int. J. Metadata Semant. Ontologies, vol. 3, no. 1, pp. 37–52, Jan. 2008.
[18] A. Harth, A. Hogan, R. Delbru, J. Umbrich, S. O'Riain, and S. Decker, "SWSE: Answers before links!," in Proc. Semant. Web Challenge, 2007, pp. 1–8.
[19] R. Valencia-García, F. García-Sánchez, D. Castellanos-Nieves, and J. T. Fernández-Breis, "OWLPath: An OWL ontology-guided query ed- itor," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 41, no. 1, pp. 121–136, Jan. 2011.