Journal of Computer Science

Original Research Paper

Performance Analysis for the Ontology based Intelligent Information Retrieval using Non Monotonic Inference Logic using SPARQL

Dr. Rajni Jindal and Alka Singhal

Department of Computer Science and Engineering, Delhi Technological University, India

Article history
Received: 01-06-2017
Revised: 16-08-2017
Accepted: 01-11-2017

Corresponding Author:
Alka Singhal
Department of Computer Science, Delhi Technological University, India
Email: alka.singhal@ipec.org.in

Abstract: The paper proposes a model for the information retrieval system (E-library) for the learner, based on his current requirements and scenario. It follows a brokerage model using non monotonic logic utilizing semantics and ontology for object description. Ontology captures the learning object properties which can help in eliminating and evaluating the usefulness of the object for a given learner. Non monotonic logic helps in inferring the current usefulness of the learning object with current requirement and rules. It will vary the results with time and person. Therefore, it can provide better user oriented search.

Keywords: E-Learning, Ontology, Information Retrieval, SPARQL, Non Monotonic Inference

Introduction

In late nineties and early twenties, Tim Berner Lee vision Semantic Web. Semantic web was envisioned as a web which can make the machine intelligent and can reduce the human effort. It was a vision which desires to make the web meaningful for machines so that if a query is thrown, the machine can capture all the user's requirements, and can serve him with the links or resources which matches his requirements without his efforts. Though there are many challenges in front of semantic web success (Paul *et al*., 2017; Molli and Skaf-Molli, 2017) like universal acceptance, sharing, security etc, but there is always a big scope to utilize the fundamentals of semantic web and make the information retrieval better and precise (Contreras *et al*., 2009).

Today's web content is suitable for human consumption. On the basis of his requirements, the user performs searches on the web, and he himself manually gather the information and draws conclusions from it. This work is not supported by software tools and extremely tedious, as the information is not machine processable. To gather the information, the search engines are becoming indispensible and most valuable tool for web users. Keyword based search engines like Google, Yahoo, Alta Vista are becoming lifelines for the web users. Though these search engines play vital role but still have various problems associated with them. Some of them are: Low and no recall, which means many times many irrelevant pages are retrieved with the bag of relevant pages, making the search list long. Sometimes, due to poor framing of queries and poor vocabulary, the search results not at all matches the user's requirements. This results in poor recall value. Results are keyword based, so synonymy and polysemy also result in poor recall value. Synonymy means that one of two or more words in the same language have the same meaning, and polysemy means that many individual words have more than one meaning (Diller, 2017). And finally for multi-requirements, we have to perform several searches to retrieve all the information and aggregate it to come conclusions.

Semantic web was vision to withstand these issues and it is becoming successful to do it. It is not a separate web, but an extension of the current one, in which information is given based on the evaluation of each link or object semantically with the users search and then depending on its relevance, results are given (Kara *et al*., 2012). Our paper is exploring semantic web benefits in the field of information retrieval in Digital libraries by providing semantic search (Ouf *et al*., 2017).

The paper suggests an e-library system which is exploring ontology at the backend, annotating database content in a RDF/XML triplet format. This will generate a connected graphical database explained later in the paper. The user query in SPARQL, a graph matching query language. The Query generates possible patterns which are matched and results are provided to the user (Bamashmoos *et al*., 2017).

The organization of the paper is as follows: Section 2 defines the overall information retrieval process explaining, the steps that are taken to drive the relevant

Science Publications

information to the user in response to his query. Section 3 describes the basic introduction of Ontology, explaining its benefits in Information retrieval process. Section 4 explains the concept of Brokerage Model, to eliminate the results which may suit learner's one requirement but not the best choice for him. Section 5 defines the basic architecture of the proposed application including its components and working. Section 6,7 describes the evaluation of the proposed model using a small dataset and also comparing it with existing keyword search. And finally concluding with Section 8, it concludes with the benefits and challenges faced by the proposed application.

Information Retrieval Process

This section deals with the basic information retrieval process. In a simple way, Information retrieval can be explained as retrieving the results in response to the user's query from a given database. More the results, match the user's requirements, better is the performance of IR (Fernández *et al*., 2011). The efficiency of the IR system is expressed in the form of precision and recall, explained later in the paper. As a black box, the whole process can be viewed as matching the user 's query with already annotated data and giving the results refer Fig. 1.

The heart of the information retrieval process lies in data representation. A system can always give the best result if it has the complete knowledge of the repository and is also able to understand the user needs. Here ontology plays a vital role. At one end, it enables the system to represent its resources by semantically annotating them and later storing them in a structured manner. On the other end, it provides machine readable learner profiling and querying the agent (SPARQL) which helps to visualize the user's current needs precisely (Li *et al*., 2016).

In the proposed paper, we are representing our database in basic RDF/XML Triples format. The RDF model is made up of triples: as such, it can be efficiently implemented and stored; other models requiring variable-length fields would require a more cumbersome implementation.

The RDF model is essentially the canonicalization of a (directed) graph, and so as such has all the advantages (and generality) of structuring information using graphs. The basic RDF model can be processed even in absence of more detailed information (an "RDF schema") on the semantics: it already allows basic inferences to take place, since it can be logically seen as a fact basis. The RDF model has the important property of being modular: the union of knowledge (directed graphs) is mapped into the union of the corresponding RDF structures; this means that: information processing can be fully parallelized in presence of partial information (an essential feature in a volatile environment like the web) the output is still a consistent RDF model, that can be successfully processed. And therefore, the final database is not a group of tables, but a graph. A graph that connects all the resources with predicate values. The user also queries in SPARQL, a graph matching query language. A query generates various patterns which are matched against the Database, and the values obtained from this process provide query results.

Ontology and its Role in Information Retrieval Process

An Ontology can be expressed as a formal, explicit specification of a shared conceptualization (Al-Yahya *et al*., 2015). It is a formal structure that provides a shared understanding of a certain domain (Blomqvist *et al*., 2017). It represents the semantics of a domain explicitly, enabling machine to retrieve data intelligently. This paper utilized the Ontology for the learning object description and the user profiling and later it is used to provide hybrid (both content and collaborative) filtering to provide user with the targeted user oriented search results. The Ontology word is quite prevalent with Information retrieval. It can be used in information retrieval system in many ways (Vesin *et al*., 2012):

- Representing Domain knowledge
- Providing metadata for key concepts and entities in the learning domain
- Allows richer description and retrieval of the learning content
- Facilitates exchange and sharing of the learning content
- Personalizing and recommending the learning content
- Designing curriculum and assessment of Learning

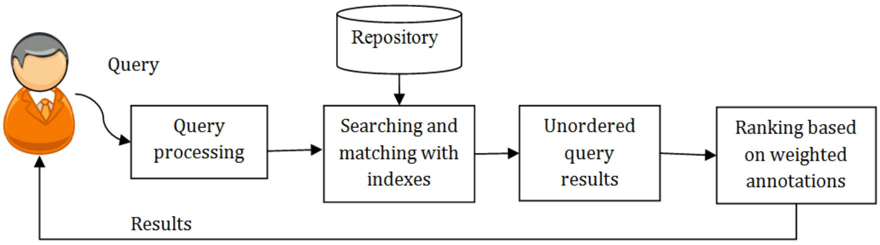
Query
Repository
Query processing
Searching and matching with indexes
Unordered query results
Ranking based on weighted annotations
Results

Fig. 1: Basic information retrieval process

Use of Non-Monotonic Rule Logic with Brokerage Model and SPARQL

In monotonic rule system, once the premises of a rule are proved, the rule is applied and its head can be accepted as the conclusion. p(x),q(y)->A. In the non-monotonic rule system, a rule may not be applied even if all premises are known because one has to consider contrary reasoning chains (Antoniou and Van Harmelen, 2004).

Our proposed system is using this concept to eliminate the results which may suit learner's one requirement but not the best choice for him. In keyword based Information Retrieval, the keywords are matched with the resources and the frequent the term is repeated in the text, higher is the rank. In such cases if the learner is not able to frame his query clear, he will get poor search results which are not relevant to him. In our proposed system, there are various attributes, which he can select to describe his needs, therefore there is no need to frame a query for multiple requirements. On the other hand, he can also prioritize the attribute depending on his present requirement. The selected attributes with priority, on the backend, is converted into SPARQL query and further processed for results. Finally, based on his selected attributes and their priority, results are shown which are precise based on his current need.

SPARQL is an RDF query language, that is, a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format. RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF (Cima *et al.*, 2017). SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware (Zhai *et al.*, 2010; Zhang *et al.*, 2019). SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graph (https://www.w3.org/TR/rdf-sparql-query/). Select, Construct, Ask and Describe are four basic variations which provide data processing and evaluation.

Working of the Model

This section discuss about the overall process of the Semantic information retrieval in the application. The proposed model is inspired by the classic model of a Search engine.

The various steps involved in Semantic Web based Information Retrieval are: (refer Fig. 2):

- The user is provided with the front End to give search queries. The attributes given to the learner are inspired by the predicates used in the context ontology used at the backend. As the system is using semantic Modeling, the query is pre-processed into SPARQL Query, which can be processed and indexed with the semantic annotations and knowledge base
- These preprocessed SPARQL Queries and searched, matched with the indexed semantic annotated Knowledge base, the documents that found after indexing with these instances are retrieved
- After matching, the unordered set of results are given. These results are mapped and ranked based on the relevancy and priority specified by the user
- After Ranking, the results are provided to the user

The various background steps which are the foundation of the application are:

- The Creation of the knowledge base: The background database is created by the semantically triplet annotated data (RDF/XML). Each resource is annotated graphically depicting triplet format (Staab, 2017). The predicates are inspired by Context Ontology adopted by the system
  The graphical structure also helps in connecting two resources sharing same predicate values. This further helps in providing a wider range of search results refer Fig. 3. Each resource in the knowledge base represents a concept and instances by means of explicit annotations
- Semantic Indexing: The main goal of the indexing module is to create description of the document. It requires the annotated data in the knowledge base which creates relation between a semantic entity and a resource. It includes Analysis and Annotation. After annotation, weightage is associated with each annotation to calculate its relevancy for a given Query. The Semantic annotations can be explained in the form of contextual ontology refer Fig. 4
- Searching; Based on Runtime queries SPARQL query, each instance is matched with query (Calvanese *et al.*, 2017)
- Ranking: Based on weightage assigned to each annotation, relevancy of each matched resources is calculated and output is provided after ranking them according to the relevancy factor

Following the example shows the working of the system. Suppose the user requires resources for Semantic Web and Web services, and it is the first time he is referring the topic.

So his query should be "Semantic Web and Web Services books for beginners".

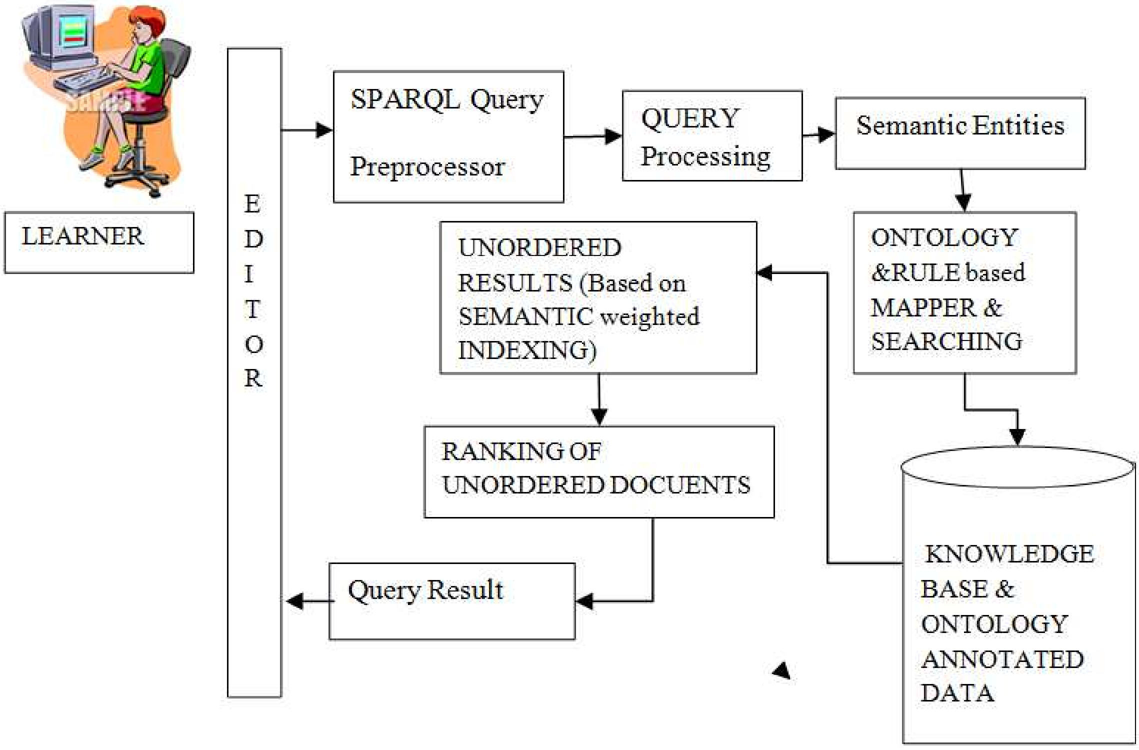
LEARNER
E D I T O R
SPARQL Query Preprocessor
QUERY Processing
Semantic Entities
ONTOLOGY &RULE based MAPPER & SEARCHING
UNORDERED RESULTS (Based on SEMANTIC weighted INDEXING)
RANKING OF UNORDERED DOCUENTS
Query Result
KNOWLEDGE BASE & ONTOLOGY ANNOTATED DATA

Fig. 2: Basic Information Retrieval Process

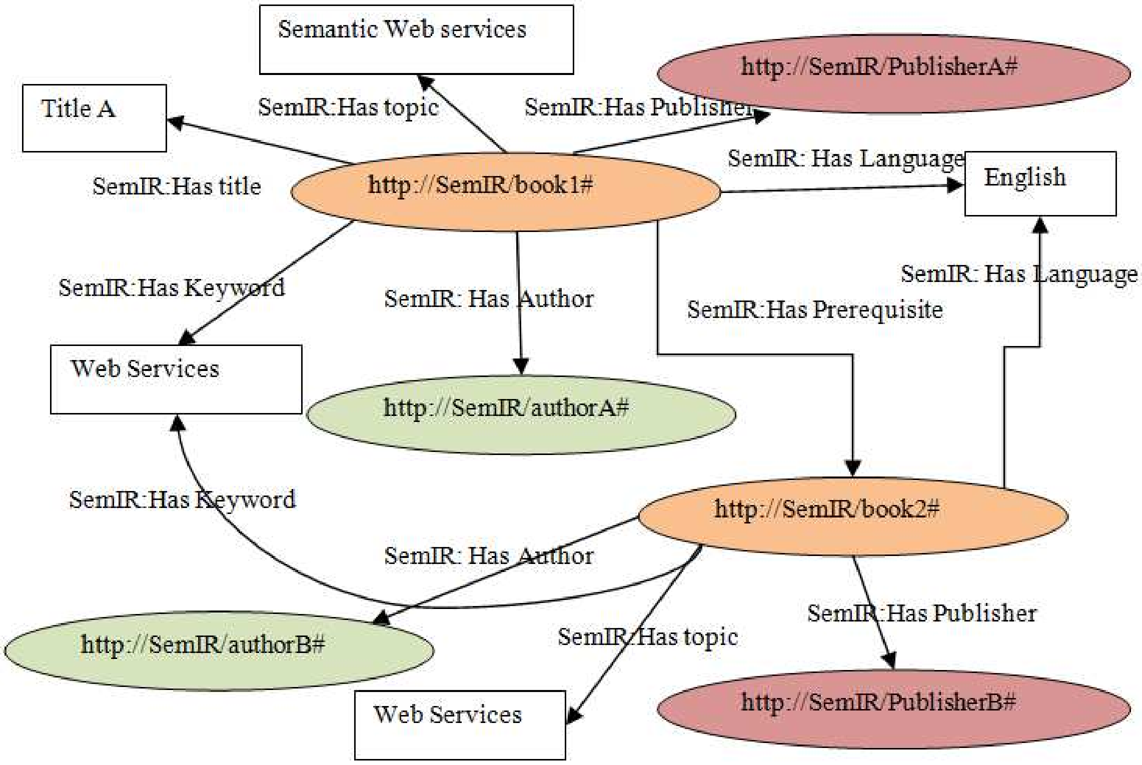

Fig. 3: The Graphical structure generated by the semantic annotation of the resources

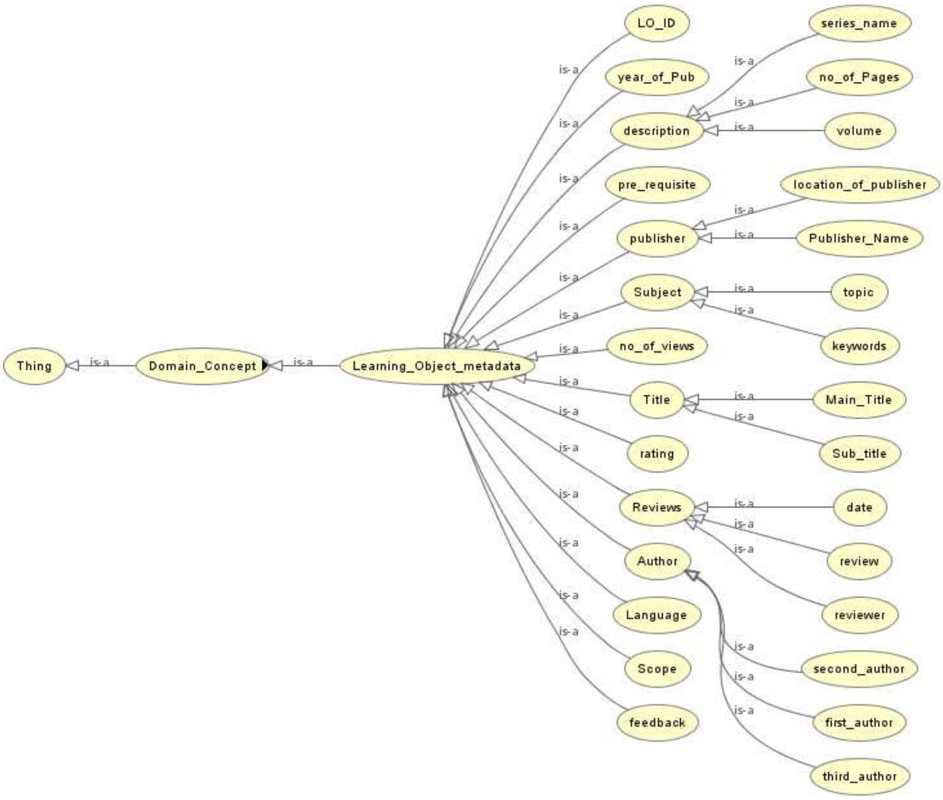

Fig. 4: Conceptualization of the learning object

Table 1: Evaluation results of the experiment queries

Query	No of Relevant Resources(X)	Total No of results shown(Y)	No of relevant resources in database (Z)	Recall (X/Y)	Precision(X/Z)
Query1	60	75	90	0.80	0.67
Query2	15	15	25	1.00	0.60
Query3	25	30	60	0.83	0.42
Query4	20	25	45	0.80	0.44
Query5	12	20	30	0.60	0.40
Query6	45	60	90	0.75	0.50
Query7	75	80	85	0.94	0.88
Query8	10	10	15	1.00	0.67
Query9	20	30	35	0.67	0.57
Query10	120	120	180	1.00	0.67
Query11	42	42	90	1.00	0.47
Query12	56	87	100	0.64	0.56
Average				0.84	0.57

As our system gives him options to select the topics and level, he selected multiple topics "Semantic Web" and "Web Services" and selected level as "beginner". If he has complete title, he can select title and write the complete name.

This Query is converted into SPARQL sub queries:

```
Prefixes SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Semantic Web and Web Services ".
?x SemIR:hastopic " Semantic Web and Web Services ".
?x SemIR:hasLevel "Beginner".
}
```

as the knowledge base is saved in XML/RDF format, it is evaluated. The SPARQL engine matches the exact matches at first level.

Rule ML

Q_Topic(X) is the X term which is selected in the topic field. Premise1

Topic(X,Y) defines X is present in Resource Y. Premise2

Level(Z,Y) defines Z is level of Resource Y. Premise3

Q_Level(Z) defines the level specified by the learner.
Then,
Rule1: Q_Topic(X),Topic(X,Y) ->Accept(Y) Priority1
Rule2: Q_Level(Z),Level(Z,Y) ->Accept(Y) Priority2

Priority helps in avoiding the clashes between the Rules.

After finishing it, it parses the query into sub queries and search in substring "Semantic Web" "Web Services" "Semantic" "Web".

Example

```
Prefixes  SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Semantic Web".}

Prefixes  SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Web Services ".

Prefixes  SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Semantic ".}

Prefixes  SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Web ".}
Prefixes  SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Semantic Web".
?x SemIR:hasLevel "Beginner".}

Prefixes  SemIR: http://www.SemIR.com/resources#
Select ?x where{
?x SemIR:hasTitle " Web Services".
?x SemIR:hasLevel "Beginner".}
```

Finally the Semantic entities which are matched and returned are ranked based on the extent they match the query and the results are shown to the user.

Evaluation Benchmark

The performance of Information Retrieval is evaluated on precision and Recall. Precision is the fraction of the retrieved documents which are relevant (Powers, 2011). It measures the extent the Retrieval process is able to understand the query and give the results which matches the user's current need to the most. Both the value range from 0 to1.

In semantic ontology based search, the query is subdivided into subqueries and predicates help to match example "Beginner" with "Level predicate" and "Semantic Web and Web services" with "topic" or "title" as compared to keyword search which will search all terms in the database without categorization giving less precision value.

Recall is the fraction of the relevant documents which has been retrieved. It measures the fraction of the all relevant documents to the search in the database which were shown to the user. It is better if none of the relevant document in the database is left from the search results.

To evaluate the performance of the system, searches were performed on a dataset of 500 resources taken from college library and annotated according to our application. The dataset was constructed containing computer science engineering related books with a proper annotation as mentioned above. Different simple queries mentioned below were fired and depth and coverage of query was increased at each level. As the database was pre-known, we calculated the value of precision and recall and plotted a graph shown Table 1 showing the success and high values attained using ontology based model.

Query1	Topic "Semantic Web and web services"
Query2	Topic "Semantic web services and web services" for Level " beginners"
Query3	Topic "Web services" Level "Beginners"
Query4	Topic "Computer network" Level "Beginners"
Query5	Topic "Linux Operating system"
Query6	Topic " Operating system"
Query7	"Data Structures"
Query8	Topic "Linux" year of publication:"2007"
Query9	"Linux for Experts"
Query10	Topic "Data structures in C".
Query11	Keyword: "Deadlock" Topic: "Operating System"
Query 12	Topic "Database management system" for Level "Experts"

In Semantic ontology based search, the precision factor is better as all the different annotations are targeted directly giving all the documents containing them, to be shown in the search result. But in the cases of "Over specialization", some documents can be overlooked which are relevant but not matching the predicates directly, so recall value was poor. Since, the main requirement was the search results should be relevant and user oriented, it can be shown that the system is working in a desired manner.

Comparative Study and Conclusion

As the database was generated from the college library itself. We were able to compare the results with the college keyword based search. With the same queries, college keyword based searching algorithm gave following results. Refer Table 2 and 3.

The results in Graph 2 and 3shows , our system perform a comparative performance and in case where the query was little twisted, it gave a considerable better result than keyword searching. Thus, we came to a conclusion that using ontology and semantic annotations, we can improve the precision value.

Query	Precision1 Semantic Search	Precision2 Keyword Based
Query1	0.80	0.64
Query2	1.00	0.55
Query3	0.83	0.63
Query4	0.80	0.57
Query5	0.60	0.43
Query6	0.75	0.66
Query7	0.94	0.66
Query8	1.00	0.67
Query9	0.67	0.7
Query10	1.00	0.66
Query11	1.00	0.75
Query12	0.64	0.7
Average	**0.80**	**0.64**

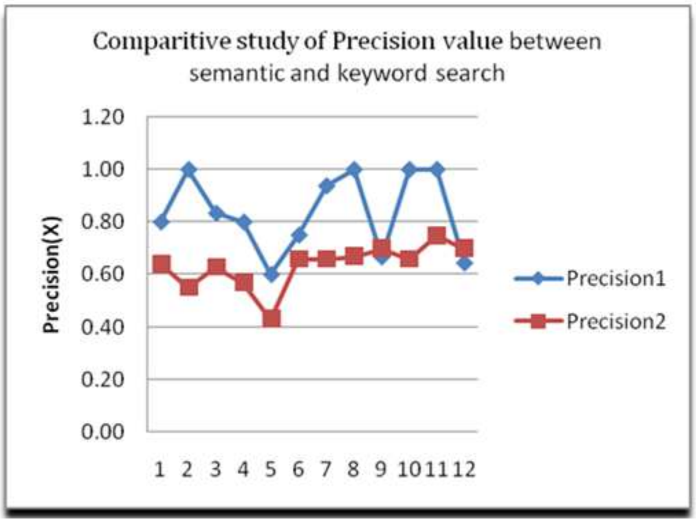

Table 2 and Graph 2: The Comparative study of Precision for Semantic and Keyword based search

Query	Recall1 Semantic Search	Recall2 Keyword Based
Query1	0.67	0.8
Query2	0.60	0.78
Query3	0.42	0.91
Query4	0.44	0.84
Query5	0.40	0.88
Query6	0.50	0.94
Query7	0.88	0.78
Query8	0.67	0.86
Query9	0.57	0.88
Query10	0.67	0.7
Query11	0.47	0.97
Query12	0.56	0.95
Average	**0.57**	**0.8575**

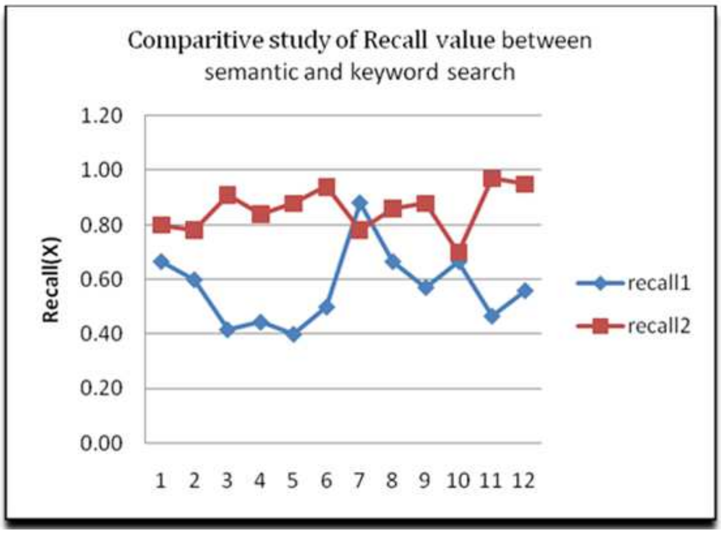

Table 3 and Graph 3: The Comparative study of Recall for Semantic and Keyword based search

Higher is precision value, more are the results relevant to the learner. Though due to overspecialization (ignoring a relevant entry due to accurate matching of all parameters mentioned in the search), recall value varies but high precision rate can be attained.

Challenges and Future Scope

A biggest challenge is the availability of semantic annotated data, as acceptance of semantic web and ontology is low in comparison to the syntactic data proliferating on the web. To accept this challenge, our application can have a broker layer which will annotate data of the resources semantically for the application. Other big challenge is Scalability, our application is presently a small scale application, which is working on a small scale of computer science resources. There will a significant need of backend, to store semantically annotated graph based data if amount of resources will increase. Poor Recall value, due to Overspecialization is also a challenge, as many time making the search accurate, some the documents containing relevant data, but not properly annotated are rejected. Our application makes many patterns for the user search and also gives him many parameters, so that the relevant data is not missed. But still this challenge has to be taken into account in future scope.

In Future scope, we can utilize the user's past usage, profiling, feedbacks and ratings to predict the results in case, learner has not provided much data in the search space.

Author's Contributions

Dr. Rajni Jindal: She has given considerable contributions to conception and design of the solution proposed in the paper and also reviewed the work critically for significant intellectual content.

Alka Singhal: She has proposed, designed and drafted the manuscript. She has acquired the data and done the Data Analysis.

Ethics

There are no ethical issues with this article.

References

Al-Yahya, M., R. George and A. Alfaries, 2015. Ontologies in E-learning: Review of the literature. Int. J. Software Eng. Appl., 9: 67-84.

Antoniou, G. and F. Van Harmelen, 2004. A semantic web primer. MIT press.

Bamashmoos, F., I. Holyer T. Tryfonas and P. Woznowski, 2017. Towards secure SPARQL queries in semantic web applications using PHP. Proceedings of the IEEE 11th International Conference on Semantic Computing, Feb. 30, IEEE Xplore Press, San Diego, pp: 276-277. DOI: 10.1109/ICSC.2017.29

Blomqvist, E., D. Maynard, A. Gangemi, R. Hoekstra and P. Hitzler *et al*., 2017. The semantic web. Proceedings of the 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, Springer.

Calvanese, D., B. Cogrel, S. Komla-Ebri, R. Kontchakov and D. Lanti *et al*., 2017. Ontop: Answering SPARQL queries over relational databases. Semantic Web, 8: 471-487. DOI: 10.3233/SW-160217

Cima, G., G. De Giacomo, M. Lenzerini and A. Poggi, 2017. On the SPARQL metamodeling semantics entailment regime for OWL 2 QL ontologies. Proceedings of the 7th International Conference on Web Intelligence, June 19-22, ACM, New York, USA. DOI: 10.1145/3102254.3102277

Diller, H.J., 2017. Corpus methods for semantics: quantitative studies in polysemy and synonymy. Int. J. Corpus Linguistics, 22: 141-151. DOI: 10.1075/ijcl.22.1.06dil

Fernández, M., I. Cantador V. López D. Vallet and P. Castells *et al*., 2011. Semantically enhanced information retrieval: An ontology-based approach. Web Semantics: Sci. Services Agents World Wide Web, 9: 434-452. DOI: 10.1016/j.websem.2010.11.003

https://www.w3.org/TR/rdf-sparql-query/

Jesús, C., O. Corcho and A. Gómez-Pérez, 2009. Six challenges for the semantic web.

Kara, S., Ö. Alan O. Cicek S. Akpınar and N.K. Cicekli *et al*., 2012. An ontology-based retrieval system using semantic indexing. Inform. Systems, 37.4: 294-305. DOI: 10.1016/j.is.2011.09.004

Li, L., S. Gao Y. Liu and X. Qin, 2016. Enhanced SPARQL-based design rationale retrieval. AI EDAM, 30: 406-423. DOI: 10.1017/S089006041600038X

Molli, P. and H. Skaf-Molli, 2017. Semantic web in the fog of browsers.

Ouf, S., M.A. Ellatif, S.E. Salama and Y. Helmy, 2017. A proposed paradigm for smart learning environment based on semantic web. Comput. Hum. Behav., 72: 796-818. DOI: 10.1016/j.chb.2016.08.030

Paul, S., A. Mitra and S. Dey, 2017. Issues and Challenges in Web Crawling for Information Extraction. In: Bio-Inspired Computing for Information Retrieval Applications, IGI Global, pp: 93-121.

Powers, D.M., 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Staab, S., 2017. Example graph for practicing SPARQL.

Vesin, B., M. Ivanović A. KlašNja-MilićEvić and Z. Budimac, 2012. Protus 2.0: Ontology-based semantic recommendation in programming tutoring system. Expert Sys. Appl., 39: 12229-12246. DOI: 10.1016/j.eswa.2012.04.052

Zhai, J. and K. Zhou, 2010. Semantic retrieval for sports information based on ontology and SPARQL. Proceedings of the International Conference of Information Science and Management Engineering, Aug. 7-8, IEEE Xplore Press, China. DOI: 10.1109/ISME.2010.79

Zhang, L., M. Zhu and W. Huang, 2009. A framework for an ontology-based e-commerce product information retrieval system. J. Comput., 4: 436-443. DOI: 10.4304/jcp.4.6.436-443