International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Semantically Enhanced Information Retrieval System

Archana P. R., Nisha T. P., Leya Elizabeth Sunny

Abstract: ***The user gives an input query in classical In-formation Retrieval (IR) system, keywords of the query are extracted and also the matching documents that contain one or more words specified by the user are retrieved. Keyword searches have a tricky time distinguishing between words that are spelled in similar way but mean something different. This often leads to hits that are completely irrelevant to the query. Se-mantic search seeks to enhance search precision by understanding searcher intent and along with the contextual significance of terms, as they seem within the searchable information space, whether on the net or within a closed system, to get more applicable outcomes. Semantically Enhanced Information Retrieval(SEIR) system can overcome the constraints of keyword based search. SEIR can semantically enhance the IR process. Therein way, searching is finished considering the meanings of query in-stead of the literal strings. Such a research automates tasks that need conceptual understanding of objects.***

Keyword: Information Retrieval; Semantically Enhanced; Ontology; SPARQL Query; User Concept.

I. INTRODUCTION

Since vocabulary is wont to describe constant meaning or concept, people will often use different words from the author or indexer of the data. So, in a very keyword based search, relevant materials are going to be missed. On the other hand, since a single word often has more than one meaning, irrelevant materials will often be retrieved. Polysemy decreases the retrieval precision by false matches, while synonymy decreases the retrieval recall by missing true conceptual matches. Limitations of keyword based search also include the inability to account for relations between search terms (e.g., "volcanoes originated in Indonesia" vs. 'volcanoes that have affected Indonesia" and "books about recommender systems" vs "systems that recommends books") or to integrate information distributed over several Web resources. So a keyword based search provides limited abilities to capture the user requirement. The user may not get the expected search results for the given query. When a user gives an input query in classical IR system, keywords of the query are extracted and the matching documents that contain one or more words specied by the user are retrieved. Keyword searches have a hard time distinguishing between words that are spelled the same way but mean something different. SEIR system can overcome the limitations of keyword based search. SEIR can semantically enhance the IR process. In that way, searching is done considering the meanings of query rather than the literal strings. Such a search automates tasks that require conceptual understanding of objects. The approach is an ontology based information retrieval system. Ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling information about individuals, their attributes, and their relationships to other individuals this can be viewed in the context of database systems. The data representation through ontology can thus semantically enhance the IR process. It takes into consideration the contextual meanings of the terms and user intentions to generate more relevant results. SEIR system can capture the concepts of user's requirements. SPARQL is a semantic query language for databases to retrieve data. Using SPARQL query, data in ontology can be accessed. The annotations of the instances or semantic relation associated with the SPARQL query is retrieved as result. But, only a domain expert knows the syntax and semantics of SPARQL query. So, SEIR has a feature to take the input from user a natural language query. Thus, the system has a translator that converts the natural language query to SPARQL query. This feature enables any naıve user to use the system. The aim is to retrieve information about a specific domain, which can be further extended to multiple domains.

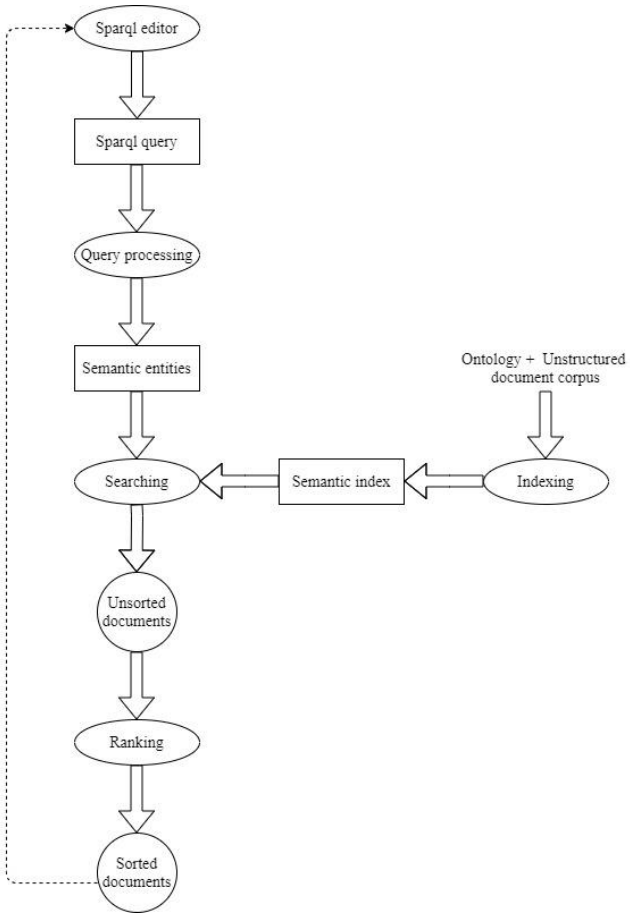
II. RELATED WORKS

Ontology based Information Retrieval [1] proposes an ontology-driven knowledge bases to improve search over large document repositories. Adaptation of classic vector space algorithm is explained along with annotation weighting and ranking algorithms. Document is represented by a document vector while an extended query vector is used. The problem of similarity between documents is solved using normalization of similarity scores. Combinational of keyword-based retrieval is adopted in order to compensate the incompleteness of knowledge base. Semantically Enhanced Information Retrieval [2] proposes searching in a massive and heterogeneous Web environment and integrates the benefits of both keyword based and semantic based search. In Semantic, the undesired effects of knowledge sparseness are minimized using web rank fusion. Conceptualizations of user needs and contents are explained using RDF.

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Archana P. R.*, Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, India. Email: archanaravindran96@gmail.com
Nisha T. P., Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, India. Email: tpnisha143@gmail.com
Leya Elizabeth Sunny, Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, India. Email:leyabijoy@gmail.com

Another work on Semantic Web based system for the domain computer science [3] proposes the development of an IRSCSD system (Information retrieval system for computer science domain) semantic web-based system for integrating ontology towards domain specific retrieval support. The proposed methodology involves four main stages. A framework for semantic web-based system is designed in the first stage. The prototype for the framework is build using Proteg´ e tool in ´ the second stage. The natural language query conversion into SPARQL query language a Python-based framework, Quepy is done in third stage. Fourth Stage involves firing of converted SPARQL queries to the ontology through Apache's Jena API to fetch the results Input to the system is natural language query. A work on effective information retrieval in semantic web [4] proposes architecture for information retrieval from semantic web that uses conceptual representations of content beyond plain keywords as knowledge bases and provides conceptual representations of user needs. This architecture handles the concept representations of the content, query extensions, matching the semantics, extraction of the relevant results in the order of relevancy. The paper also presents a discussion of algorithms used in the different components of the proposed semantic search architecture. A related work which is on ontology based information retrieval [5] proposes an ontology-based semantic retrieval scheme for knowledge search and retrieval from domain documents. The basics steps for ontology based semantic retrieval for engineering domain are, first the domain ontology is constructed using the graph approach. GRAONTO is a graph-based approach for automatic construction of domain ontology. The next step is to discover the latent relationship between queries and ontology semantic features. For doing this step adopt query semantic graph are used to represent query. For semantic retrieval a graph based k-mean method is proposed to partition the domain documents into clusters. Another approach on ontology based similarity for product information retrieval [6] proposes the idea of using ontologies to produce similarity measures that are based on semantics A Case Based Reasoning system is consist of domain knowledge, a case base, and a search mechanism based on a similarity measure. CBR uses similarity measures to identify cases which are more relevant to the problem to be solved. Formal concept analysis (FCA) is a data processing method that can be used to design ontologies. In this paper, FCA along with a theoretical framework is proposed for developing product and process ontologies. Most semantic similarities are defined in terms of the number of edges between the classes that they compare (edge-counting similarity measures). Semantically enhanced information retrieval [7] approaches the use of both semantic information and context in the information retrieval process with the goal of developing context-based semantically enhanced information retrieval systems. The three types of context information relevant for an IR system knowledge context information, user context information and constraint context information. SOBA is a system that create a soccer specific knowledge base from heterogeneous sources. The information retrieval model is tested using the ArhiNet system an integrated information retrieval system for archive content, based on semantic enhancements. ArhiNet is an IR system for developing, managing and retrieving archive content based on semantic enhancements.Wordnet based information retrieval [8] proposes information retrieval based on Wordnet ontology. The semantic terms in a text document is identified using Word Sense Disambiguation [WSD]. Annotation of a document with semantic terms is done with RDF graphs or triples. KIM is a tool used to give automatic annotation with reference to the ontologies. Polysemy and synonymy are considered while using WSD. A research on Jena page ontology extracting and processing [9] aims at solving the problems in web ontology extraction and processing like unified ontology language, lacking of tools, limited tools plain functional method. Jena is an open source semantic web tool based on Java, a Java class library. It contains interfaces for representing models, resources, properties, literal, statements and other key concepts of RDF. Ontology parser of Jena includes, parsing of RDF, the query support for RDQL, and the parsing of the OWL. The proposed system also deals with conversion from natural language to SPARQL. A work on ontology semantic tree based natural language approach [10] proposes a natural language interface which accepts natural language as input and generates SPARQL queries as output. Ontology semantic tree is used to represent semantic conceptual structure transformed from dependency tree. In semantic tree, each node except root corresponds to concept of class; edges correspond to concept of relation. Input NL query is subjected to word segmentation and POS tagging and produce dependency tree using Open Source toolkit CRF++. Dependency tree is transformed to semantic tree with support of ontology. SPARQL query is generated based on this tree and used for retrieving answer. A work which maps Natural language questions to SPARQL queries [11] presents a model named Sem-Qas that dynamically forms complex SPARQL queries in Job search domain by combining the triple patterns generated for atomic filtering constraints. There are mainly three modules. They are Semantic Linguistic Analysis, Query Template Matching, and SPARQL Query Generation. In first module, the errors like grammatical and spelling mistakes, incomplete questions, and use of jargons are handled. The second module is implemented using pattern matching. A detailed survey on SPARQL query composition [12] presents SparqlFilterFlow, a novel approach for visual SPARQL querying based on the lter / ow model. SparqlFilterFlow can be used for the construction of complex query expressions. The directed acyclic graphs visualize expressions, where the nodes dene the lter criteria and the edges depict the ow of data. SparqlFilterFlow enables the composition of SPARQL queries. It can be applied to any SPARQL endpoint and allows for creating complex SPARQL queries with only little training. Another work is Semantic search in digital library [13] discuss about ongoing research in developing semantic digital library in academic institutions. The paper explains limitations of traditional search engines in dealing with domain knowledge, relationship among terms etc.

So they proposed an ontology based retrieval model meant for utilization of complete domain ontologies and KB's which support semantic search in digital library. A work which deals with query answering over ontologies [14] explains that Query answering can be used by users and applications to interact with ontologies and data. SPARQL is a query language supported by most RDF triple stores. Query evaluation method explained by this paper is based on sub graph matching. SPARQL includes several entailment regimes, including RDFS and OWL. The algorithm proposed in the paper extends techniques used for conjunctive query answering and propose a range of novel optimizations in particular for SPARQL-OWL queries. The system proposed combines existing tools such as ARQ, the OWL API and the Hermit OWL reasoner to implement an algorithm that evaluates basic graph patterns under OWL's Direct Semantics Some of the works mentioned only deals with information retrieval from ontology. But a naıve user cannot directly query into ontology. The system being proposed integrates querying from ontology with conversion of natural language to SPARQL. This makes the ontology based querying easier for a naıve user.



III. SYSTEM ARCHITECTURE

A. Proposed Method

As in fig 1, the system consists of the following processes. First, the input is given as a SPARQL query. This is executed against the Knowledge Base which returns the list of semantic entities that satisfy the query (based on exact match).The query execution returns a set of tuples that satisfy SPARQL query. Semantic entities are extracted from tuples. Annotated documents with above instance are retrieved and presented (based on approximate match) picted in the form of graphs.

Only an expert user who knows the syntax and semantics of the SPARQL query language will be able to query from such a system. So a system where user can input a natural language query through a natural language interface is proposed. The proposed system translates the natural language query into a SPARQL query. So the system can be used by any naive user.

IV. IMPLEMENTATION

1) Commands for Protege Open the terminal and point it to the Protege folder.

/Protege-4.3. /run.sh

2) Commands for Quepy

- Starting a Quepy Project
  quepy.py startapp ¡appname¿
- Running a Quepy project
  python main.py

3) Commands for PyQt

The user designed interface is stored as .ui file. This .ui file is converted to .py file by giving the command:

pyuic4 -x Filename.ui -o Filename.py

The following figures show the results of the semantically enhanced information retrieval. User Interface is shown in figure 2. PyQt4 is used for building the natural language interface. The interface contains a LineEdit to accept the user query, a push button to process the input query. It also contains textBrowser to display the translated SPARQL query and the retrieved results.PyQt4 generates a .ui file of the interface created.

Fig. 2. User Interface

This .ui file is converted to .py file by giving the following command: *pyuic4 -x Filename.ui -o Filename.py* Following is a code snippet from the interface file:

```
from PyQt4.Qt import *
from PyQt4 import QtCore, QtGui
self.pushButton = QtGui.QPushButton(Dialog)#to create a button
self.pushButton.setGeometry(QtCore.QRect(500, 140, 98, 27))
self.pushButton.clicked.connect(self.handleButton)      #to handle the button
defhandleButton(self):
question=self.LineEdit.text()
query,result=m.findquery(question)
print query
for i in result:
for j in i:
self.textBrowser_2.append(j)
print
self.textBrowser.setText(query)
```

Translation, processing of query involves: Tokenization, POS tagging, parsing, Intent determination. Then, the translation of processed natural language to SPARQL query is done using Quepy. Prefixes required for execution are also added. Following is a code snippet for the processing of natural
language query:

```
tokenizer = EnglishTokenizer()
tagger = EntityTagger(trie, tokenizer)
parser = Parser(tokenizer, tagger)
engine = IntentDeterminationEngine() Following prefixes are
added:
PREFIX owl: <http://www.w3.org/2002/07/owl>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf>
PREFIX foaf: http://xmlns.com/foaf/0.1
PREFIX skos: <http://www.w3.org/2004/02/skos/core>
PREFIX quepy:<http://www.machinalis.com/quepy>
PREFIX ker:http://www.semanticweb.org/user/ontologies
/2017/0/untitle ontology-33 prefix of the built ontology
'kerala. owl'
```

Following is a code snippet of the translation of a type of query:

```
import quepy
target, query, metadata = ontology.getquery(question)
/*'question'
is parsed and tokenized*/
class Person(Particle):
regex = Plus(Pos("NN") | Pos("NNS") |Pos("NNP")
|Pos("NNPS"))
def interpret(self, match):
name = match.words.tokens
return IsPerson() + HasKeyword(name)
regex = Lemma("Who")+Lemma("be")+ Person() +
Question(Pos("."))
/*regular expression for the type of question "Who is M M
Mani?"*/
```

Here, ontology is the directory holding the files for the whole query processing in quepy. It contains the python files that are created to handle each type of query. These python files contain the regular expression for the particular query. It also has a file dsl.py that handles the classes and semantic relations to be accessed by each type of query. Following is the code snippet in dsl.py to handle a query that has to access the semantic relations for its result:

```
from quepy.dsl import FixedType, HasKeyword,
FixedRelation,
FixedDataRelation
HasKeyword.language = "xsd:string"
HasKeyword.relation="rdfs:label"
class CollectorOf(FixedRelation):
for the query like "Who is the collector of Kozhikode?"
relation = "ker:hasCollector"
reverse = True
```

Following is a code snippet to handle a query accessing the class instances for its result:

```
class IsPerson(FixedType):
fixedtype = "ker:people"
```

- Building ontology: Ontology of the domain under concern is built using Protege. Protege is opened by running. /run.sh file in the Prot´eg´e directory. The list of concepts (classes), object properties, data properties and instances of each class are listed. Ontology for the domain Kerala state was built.
- Execution: The translated SPARQL query is executed against the ontology. Following code snippet is used for this:

```
=rdflib.Graph()
g.parse(filename) filename is the built ontology
ontology = quepy.install("ontology") /*ontology is the
directory
holding the files
for the whole query processing in quepy*/
result = g.query(query)
```

- Results: The SPARQL query and retrieved results is presented to user through the natural language interface. Following code snippet is used for this:

```
for i in result:
for j in i:
self.textBrowser2.append(j)
print
self.textBrowser.setText(query)
```

- Queries: Queries can be mainly divided into two types depending on its complexity. They are simple queries and complex queries. Simple queries are those which access only a single instance. Complex queries are those which access multiple instances and give the result as a conjunction of all the instances. Example of a simple query is "Who is Person?" (Query no.1 mentioned below). Example of a complex query is "Which department is handled by Person?" (Query no.5 mentioned below). Following are the example of queries shown in fig 3,4 and 5:

1. Who is MM Mani?

```
class Person(Particle):
regex = Plus(Pos("NN") | Pos("NNS") |Pos("NNP")|
Pos("NNPS"))
def interpret(self, match):
name = match.words.tokens
returnIsPerson() + HasKeyword(name)
regex = Lemma("Who")+Lemma("be")+ Person() +
Question(Pos("."))) /
*regular expression for the type of question "Who is M M
Mani?"*/
```

2. Which department is handled by M M Mani?

```
class Department(Particle):
regex = Plus(Pos("DT")| Pos("NN") |Pos("NNS")|
Pos("NNP") |Pos("NNPS"))
def interpret(self, match):
name=match.words.tokens.title()
IsDepartment()+HasKeyword(name)
regex Lemma("Which")+Lemma("department")+Lemma
("is")+Lemma("handled
by")+Person()+Question(Pos("."))
/*regular expression for the question "Which department is
handled by M M Mani?" */
```

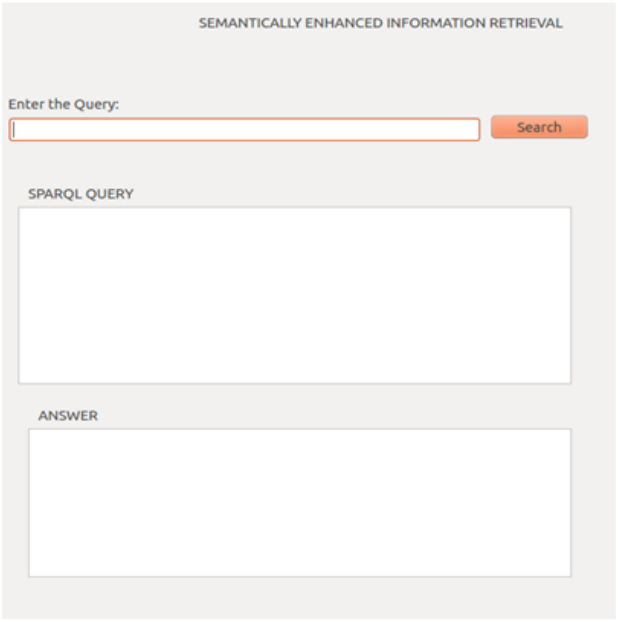


Fig. 2. User Interface

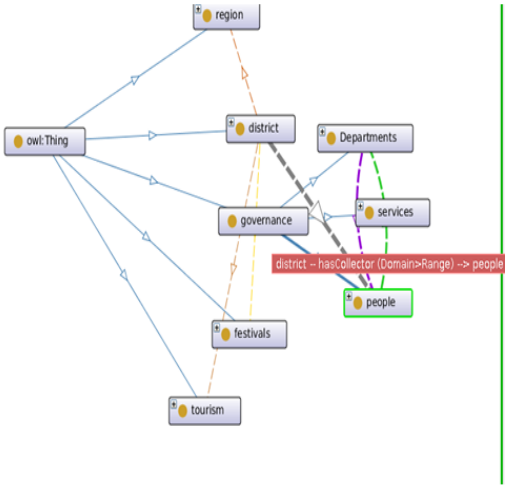


Fig. 3. Kerala.owl

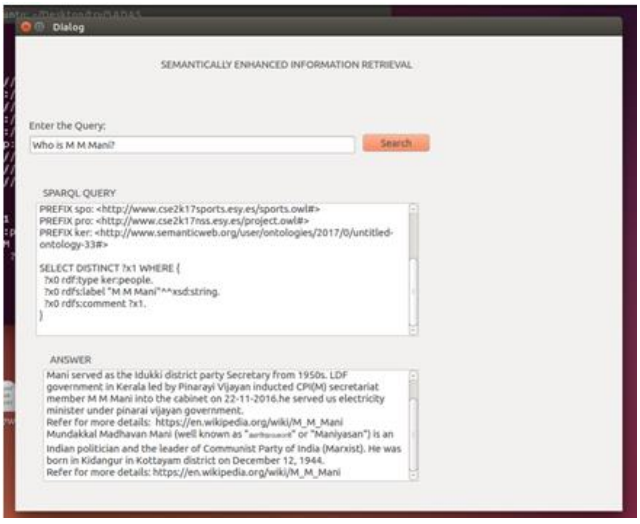


Fig 4. Example Query 1

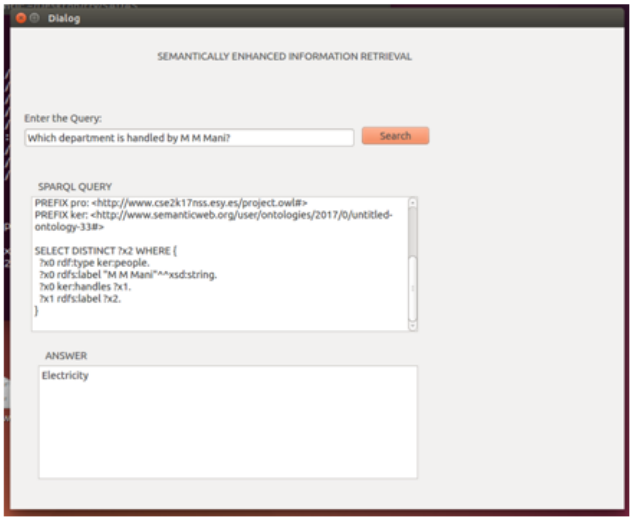


Fig.5. Example Query 2

V. ANALYSIS

The reliability and validity of the project is estimated through series of evaluation processes using certain evaluating measures. These tests help in reckoning the flaws and performance of the system.

Accuracy: This feature tells how close the result obtained from querying is with the actual result. Various queries were run against the ontology and the results are observed to be accurate. Even though false results are not generated, the queries for which the regular expressions are not defined will not yield a result.

Precision-Recall: Precision-Recall is a major evaluating factor of Information Retrieval. Here, precision shows the result relevancy and recall is the amount of how many result are truly returned. The result obtained is accurate and precise to an extent. A well detailed result is seldom observed owing to the limitation of ontology.

$$\text{precision} = \frac{\text{relevant document t}\cap\text{ retrieved documents}}{\text{retrieved documents}} \quad (1)$$

$$\text{recall} = \frac{\text{relevant document }\cap\text{retrieved documents}}{\text{relevant documents}} \quad (2)$$

Performance: Performance of the system in calculated by comparing the two basic operations (ie. NL to sparql conversion and sparql to result) with that of existing system such as query.wikidata.org .The time taken to generate the SPARQL query for similar question is compared and the performance is calculated.

Dataset Description: The ontology being used is based on the domain Kerala State. It has various classes such as: District, Governance, Festivals, Region, Tourism. The class governance has got 3 subclasses such as: Department, People, and Services. Individuals of each class are given in the ontology.

Table 1 shows the semantic relations in the ontology. The various data properties are: deptName, dist Name, festival NAME, fest Name, Headquaters, Name, region name, servicename and tourist Place Name.

Result Analysis: Various factors are taken into consideration for analysis of the performance, reliability and validity of the system. The results are depicted in the form of graphs. Depicts how the ontology system trumps over the traditional database system by inferring meanings from the available data.Fig 6. Advantage of Ontology system over Database system

This adds semantics to the system and the information at hand and we can see from the graph that with adequate relations the inferred triples will increase exponentially.

Fig 7 represents the performance of the system in comparison with the performance of system in query.wikidata.org. Clearly, we can see the time taken to obtain results from the ontology is lesser in the project system and the gap is increasing.

Table I : Semantic relations in the ontology

Domain	*Relation*	*Range*
District	belongs to	Region
Department	handled by	People
People	Handles	Department
District	Hascollector	People
District	Hasfestival	Festivals
District	hasTouristplaces	Tourism
People	isCollectorOf	District
Tourism	isinDistrict	Places

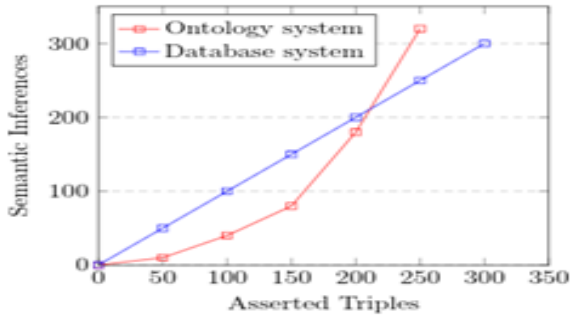


Fig 6. Advantage of Ontology system over Database system

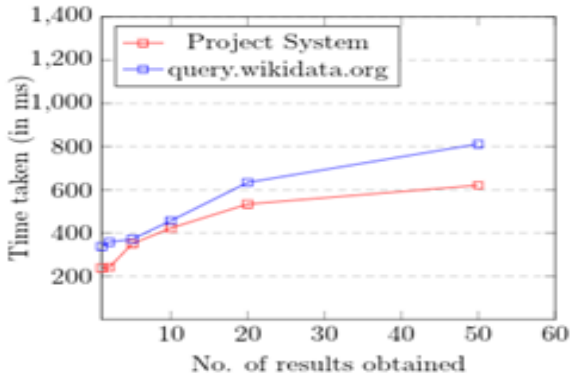


Fig 7. Performance of SEIR system and query.wikidata.org

VI. DISCUSSIONS

Feasibility study is an analysis of the viability of the proposed idea. It is an assessment of practicality of proposed project and ensures that the project is legally and technically feasible and economically justifiable. All the projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of a project at earliest possible time. The two criteria to judge are cost required and value to be attained. Thus, a historical background of project, a description of the service, details of operational management, marketing research and policies are provided. In this, theoretical background of feasibility analysis for projects in the field of semantic web with particular emphasis on the theory and application of information retrieval is provided. Generally feasibility studies technical development and project implementation.

1. Technical Feasibility

The technical feasibility assessment is focused on gaining an understanding of the present technical resources for the project and their applicability to the expected meets of the proposed system. It is the evaluation of the hardware and the software and how it meets the needs of the proposed system The proposed methodology can be implemented using programming languages Python and SPARQL. Ontology can be generated using Protege 4.3. NLP is done by giving regular expression in Python. All mentioned technologies are widely used in the world and available for free use. So the entire project is found technically feasible.

2. Economic Feasibility

In economic feasibility, the positive economic benefits of proposed system are determined. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis. The proposed methodology requires no explicit cost. The technologies mentioned above are found freely accessible and does not require any registration or purchase of any kind. So the project is economically feasible.

3. Cultural Feasibility

It is a measure of how the end users feel about the proposed system .The results of the project does not clash with any enterprise's culture and it does not cause any environmental harm to the local and general culture. This project is user friendly.

4. Legal Feasibility

The proposed system does not conflict with legal requirements or violate any rules and hence is legally feasible.

5. Schedule Feasibility

This is probability of project to be completed within its scheduled time limits. The aim is to complete the project in given period of time and measure how reasonable the project is.

VII. CONCLUSION

The proposed system retrieves the information by semantic enhancement through an ontology based approach.

User is able to retrieve the expected result. The proposed system is a search engine where information retrieval takes place efficiently. A naive user can give the query easily in natural language, which makes this system user friendly. A semantics based search provides good capabilities to capture the concept of the user requirement.

This system aims to work for a particular domain which can be further extended to entire knowledge base of different domains. As future work, a system which can integrate various ontologies together can be designed. Here ranking of the retrieved content can be done in their order of relevance. Word Net, a lexical database for English language can also be imported in the system to enhance the query.

REFERENCES

1. Castells, Pablo, Miriam Fernandez, and David Vallet. "An adaptation of the vector-space model for ontology-based information retrieval." IEEE transactions on knowledge and data engineering 19.2 (2007).
2. Fern´andez, Miriam,"Semantically enhanced information retrieval: An ontology-based approach." Web semantics: Science, services and agents on the world wide web 9.4 (2011): 434-452.
3. Bansal, Ritika, and SonalChawla. "Design and development of semantic web-based system for computer science domain-specific information retrieval." Perspectives in Science 8 (2016): 330-333
4. Thangaraj, Muthuraman, and G. Sujatha. "An architectural design for effective information retrieval in semantic web." Expert Systems with Applications 41.18 (2014): 8225-8233.
5. Zhang, Xutang, "Ontology-based semantic retrieval for engineering domain knowledge." Neurocomputing 116 (2013): 382-391
6. Akmal, Suriati, Li-Hsing Shih, and Rafael Batres. "Ontology-based similarity for product information retrieval." Computers in Industry 65.1 (2014): 91-107.
7. Cioara, Tudor,"A context-based semantically enhanced information retrieval model." Intelligent Computer Communication and Processing, 2009.ICCP 2009.IEEE 5th International Conference on.IEEE, 2009.
8. Fellbaum, Christiane. WordNet. Blackwell Publishing Ltd, 1998.
9. Min, Wang, "The research on the jena-based web page ontology extracting and processing." Semantics, Knowledge and Grid, 2005.SKG'05.First International Conference on.IEEE, 2005.
10. Li, Shusen, Zhiyang He, and Ji Wu. "An ontology semantic tree based natural language interface." Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on. IEEE, 2014.
11. Karim, Naila, "Mapping Natural Language Questions to SPARQL Queries for Job Search." Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on. IEEE, 2013.
12. Haag, Florian, Steffen Lohmann, and Thomas Ertl. "SparqlFilterFlow: sSpringer International Publishing, 2014.
13. Jensen, John R. Introductory digital image processing: a remote sensing perspective. Univ. of South Carolina, Columbus, 1986.
14. Kara, Soner, "An ontology-based retrieval system using semantic indexing." Information Systems 37.4 (2012): 294-305.
15. Gruber, Thomas R. "A translation approach to portable ontology specifications." Knowledge acquisition 5.2 (1993): 199-220.
16. Fern´andez-L´opez, Mariano. "Overview of methodologies for building ontologies." (1999).
17. www.protege.stanford.edu
18. http://quepy.readthedocs.io/en/latest/tutorial.html
19. https://wiki.python.org/moin/PyQt

AUTHORS PROFILE

Archana P. R., received Bachelor of Technology in Computer Science and Engineering from NSS College of Engineering, Palakkad in 2017 and currently pursuing Master of Technology in Computer Science and Engineering from Mar Athanasius College of Engineering, Kothamangalam affiliated to APJ Abdul Kalam Technological University. Her research interest is NLP, Semantic web and cloud computing.

Nisha T. P., received Bachelor of Technology in Computer Science and Engineering from Thejus Engineering College, Vellarakkad in 2017 and currently pursuing Master of Technology in Computer Science and Engineering from Mar Athanasius College of Engineering, Kothamangalam affiliated to APJ Abdul Kalam Technological University. Her research interest is in Deep Learning and Data Mining.

Leya Elizabeth Sunny, is currently working as assistant professor in the Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India. She received her B-Tech Degree in 2004 in Computer Science and Engineering from MG University and M-Tech in 2011 in Information System Security. She has around 12 years of teaching and industrial experience. She is interested in the area of Data Security and Cryptography.