Volume 2, No. 4, April 2011

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

SEMANTIC RETRIEVAL FOR HOMONYMS USING RDF AND SPARQL

Amrita Bhandari*[1], Shalini Batra[2]

*[1] Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India
amrita.bhandari.it@gmail.com,
[2] Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India
sbatra@thapar.edu

Abstract: With the advancement of internet and its use becoming global, users are finding it difficult to retrieve results for synonyms and homonyms. To overcome this upcoming problem, semantics methods for retrieval of required data are being studied. This research paper explores the methodology for semantic based retrieval of Homonyms - words with similar meanings, through the use of similarity measures like RDF, OWL, etc. After thorough analysis it was realized that Resource Description Framework (RDF), is an effective tool for Information Retrieval as it can be easily implemented and stored. Twinkle SparQl tool has been used for querying the data stored using RDF format. Firstly an example Database has been built using RDF, which is of less fine granularity and then a semantic retrieval for it using SparQl language has been represented.

Keywords: RDF, Homonyms, SparQl, Semantic Web

INTRODUCTION

With the advent of Internet as a medium for sharing resources and its global rapid development, it has emerged as a huge repository of unstructured data. This has made the searching and knowledge extraction a very cumbersome task. The traditional retrieval techniques which involve searching through the index, content, keywords, etc. are not efficient for retrieving the homonyms and thus came out with many problems [1], one of which is retrieval of results for homonyms. The user does not get precise results. One solution to this problem is use of Resource Description Framework (RDF) [2].

RDF is used to represent information modeled as "graph": a set of individual objects, along with the set of connections among those objects [3]. The ways to query a RDF file containing the data for a homonym has been presented in this paper.

There are set of homonyms like eclipse, which has different meanings, one in context of astrology representing Solar or lunar eclipse, one in context of car, eclipse is also a name given to software, one is the name of a movie, and one eclipse is a novel. So we need to find out the methods to deliver the exact results to the user which he is interested in. In other words, such a framework would need to be based on metadata (data about data) that describes content of Web resources [4]. This is achieved by adding Annotations to the data.

Annotations are viewed as statements made by an author about a Web document [5]. Annotations, also described as metadata, allows the author to add properties about some given content for his web document. The semantic annotation of texts consists of extracting semantic relations between domain relevant terms in texts. The annotations have therefore been added to the RDF repository.

This paper concentrates on describing the RDF infrastructure of a homonym – Eclipse, additions of annotation to it, and its implementation using the Twinkle SparQl Tool.

RELATED WORK

McEnergy and Wilson proposed that an annotated corpus may be considered to be a repository of linguistic information made explicit through concrete annotation [6]. The benefit of such an annotation is clear: it makes retrieving and analysing information about what is contained in the corpus quicker and easier. This work of adding annotations to the repository is retrieved using a Query Language. For efficient querying, SPARQL extensions allowing the user to query the Semantic Web with preferences, was proposed by Siberski [7]. New keywords like PREFERRING, CASCADE are added to the SPARQL grammar in order to favor query answers which match user-defined preference criteria. The answers independent of others are returned to the user.

Further extension to this is, Networked Graphs, proposed by Schenk and Staab [11], which allows the users to define RDF graphs not only by extensionally listing content but also by using views on other graphs. They showed that Networked Graphs allow for defining, exchanging and executing SPARQL rules, SPARQL views and RDF data integration in a decentralized fashion.

For optimization of the retrieval and the query results, Ruckhaus [8] proposed the estimation of cost and cardinality of individual query predicates based on selectivity estimations. To reduce the complexity of Semantic Web queries, Query optimization strategies have been developed and with this, the run time performance for results retrieval also increased.

SEMANTIC WEB

According to Tim Berners-Lee “The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation”[13]. The vision of the Semantic Web is an extension of the existing Web through which machines are able to interoperate and work on our behalf. .It requires resource description so that machines know what they mean (metadata) [14].

NEED OF RDF FOR HOMONYMS

For words like Eclipse, Trident, Swift, Twinkle, Gate etc which with same spelling have different meaning in different contexts, optimization of the search is required, so that user gets the accurate results. For example if the user enters word swift, he gets results corresponding to:

1. Society for Worldwide Interbank Financial Telecommunication
2. Swift CAR
3. A reel or turning instrument, for winding yarn, thread, etc.
4. A variety of potato

So to limit the search and give the user accurate results, annotations to the data have been added in the RDF file. By doing this the outcome of unnecessary results have been reduced.

RESOURCE DESCRIPTION FRAMEWORK (RDF)

RDF can be defined in three simple rules:

1. A fact is expressed as a triple of the form (Subject, Predicate, and Object). It's like a little English sentence.
2. Subjects, predicates, and objects are names for entities, whether concrete or abstract, in the real world. Names are either 1) global and refer to the same entity in any RDF document in which they appear, or 2) local, and the entity it refers to cannot be directly referred to outside of the RDF document.
3. Objects can also be text values, called literal values.

A simple RDF assertion is shown on “Fig. 1” below.

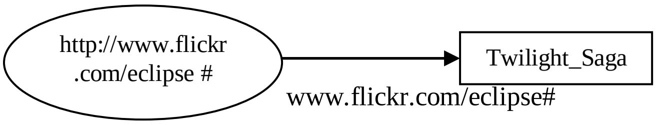


Figure 1: A simple RDF assertion

Representing URIs

In work with RDF, URI’s are abbreviated in several ways, using: namespace, PREFIX and ENTITY definitions, depending on the context:
xmlns: lib = “http://some.host.edu/directory”
or, PREFIX <lib:http://some.host.edu/directory>
or, !ENTITY lib“http://some.host.edu/directory”
If the namespace abbreviation for “eclipse” is substituted for each occurrence of “eclipse:” in the data encoding using XML entities above, then, <eclipse: type>Car</eclipse: type>, is actually being represented as:
< http://www.flickr.com/eclipse#type>
Car
</ http://www.flickr.com/eclipse #type>

Representing Properties

1) *Properties encoded as XML entities*

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf: RDF xmlns:
rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns: eclipse="http://www.flickr.com/eclipse#">
<rdf: Description rdf:
about="http://en.wikipedia.org/wiki/Mitsubishi_Eclip
se">
<eclipse: type>Car</eclipse: type>
<eclipse: brand>Mistubishi</eclipse: brand>
<eclipse: meaning>Racehorse</eclipse: meaning>
<eclipse: info>car</eclipse: info>
</rdf: Description>
```

2) *Properties encoded as XML attributes*

```
<? xml version="1.0" encoding="UTF-8"?>
<rdf: RDF xmlns:
rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns: eclipse="http://www.flickr.com/eclipse#">
<rdf: Description rdf:
about="http://en.wikipedia.org/wiki/Mitsubishi_Eclip
se">
eclipse: type= “Car”
eclipse: brand= “Mistubishi”
eclipse: meaning= “Racehorse”
eclipse: info= “car”
</rdf: Description>
```

RDF Model of an annotation

The RDF model representing the annotations is given in the “Fig. 2”.

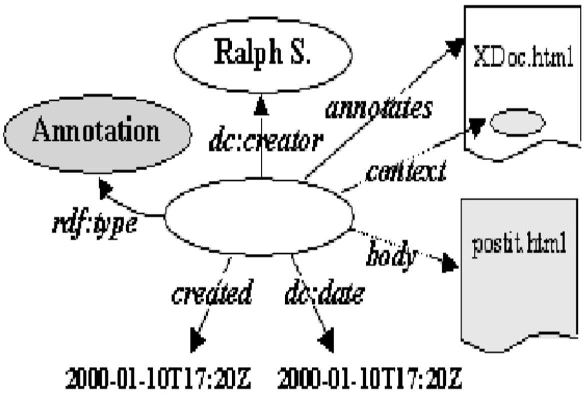
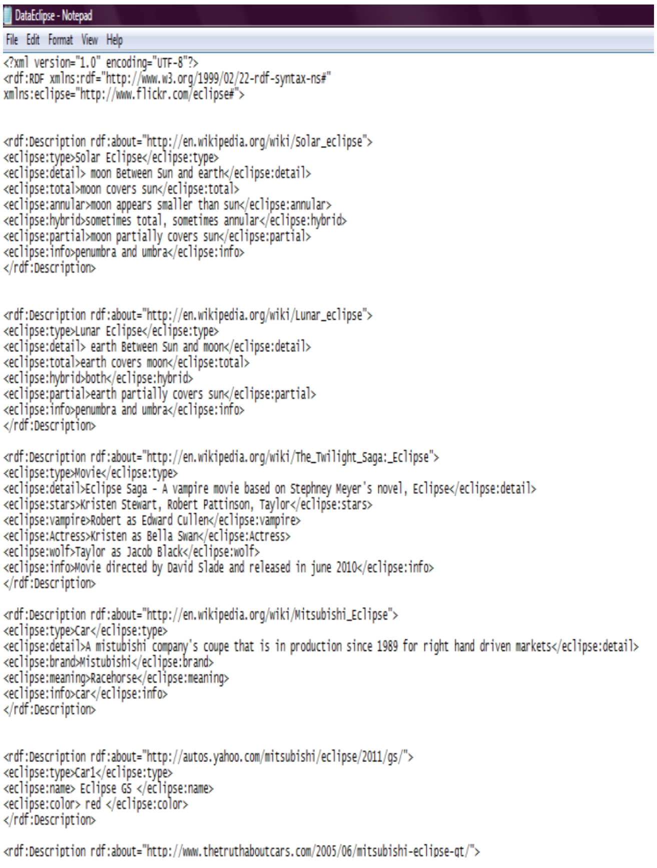


Figure 2: The RDF model of an annotation [5].

RDF Documentation Creation:

RDF Editor or notepad can be used to make the RDF document for “eclipse” repository. Self defined tags are used to annotate the various results. “Fig. 3” is example repository.

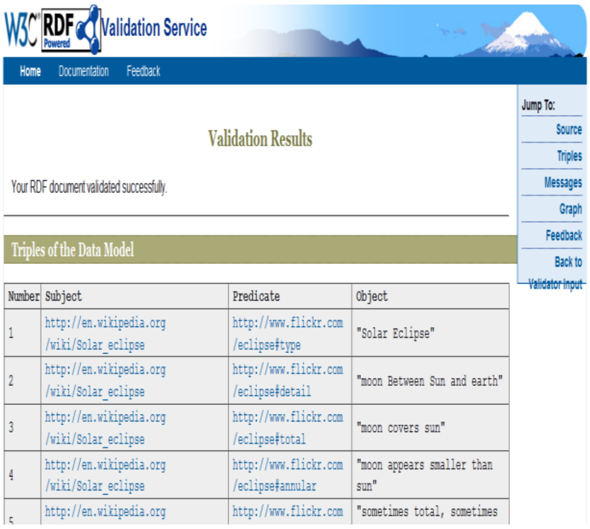


```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:eclipse="http://www.flickr.com/eclipse#">


<rdf:Description rdf:about="http://en.wikipedia.org/wiki/Solar_eclipse">
<eclipse:type>Solar Eclipse</eclipse:type>
<eclipse:detail> moon Between Sun and earth</eclipse:detail>
<eclipse:total>moon covers sun</eclipse:total>
<eclipse:annular>moon appears smaller than sun</eclipse:annular>
<eclipse:hybrid>sometimes total, sometimes annular</eclipse:hybrid>
<eclipse:partial>moon partially covers sun</eclipse:partial>
<eclipse:info>penumbra and umbra</eclipse:info>
</rdf:Description>


<rdf:Description rdf:about="http://en.wikipedia.org/wiki/Lunar_eclipse">
<eclipse:type>Lunar Eclipse</eclipse:type>
<eclipse:detail> earth Between Sun and moon</eclipse:detail>
<eclipse:total>earth covers moon</eclipse:total>
<eclipse:hybrid>both</eclipse:hybrid>
<eclipse:partial>earth partially covers sun</eclipse:partial>
<eclipse:info>penumbra and umbra</eclipse:info>
</rdf:Description>

<rdf:Description rdf:about="http://en.wikipedia.org/wiki/The_Twilight_Saga:_Eclipse">
<eclipse:type>Movie</eclipse:type>
<eclipse:detail>Eclipse Saga - A vampire movie based on Stephney Meyer's novel, Eclipse</eclipse:detail>
<eclipse:stars>Kristen Stewart, Robert Pattinson, Taylor</eclipse:stars>
<eclipse:vampire>Robert as Edward Cullen</eclipse:vampire>
<eclipse:Actress>Kristen as Bella Swan</eclipse:Actress>
<eclipse:wolf>Taylor as Jacob Black</eclipse:wolf>
<eclipse:info>Movie directed by David Slade and released in june 2010</eclipse:info>
</rdf:Description>

<rdf:Description rdf:about="http://en.wikipedia.org/wiki/Mitsubishi_Eclipse">
<eclipse:type>Car</eclipse:type>
<eclipse:detail>A mistubishi company's coupe that is in production since 1989 for right hand driven markets</eclipse:detail>
<eclipse:brand>Mistubishi</eclipse:brand>
<eclipse:meaning>Racehorse</eclipse:meaning>
<eclipse:info>car</eclipse:info>
</rdf:Description>


<rdf:Description rdf:about="http://autos.yahoo.com/mitsubishi/eclipse/2011/gs/">
<eclipse:type>Car1</eclipse:type>
<eclipse:name> Eclipse GS </eclipse:name>
<eclipse:color> red </eclipse:color>
</rdf:Description>

<rdf:Description rdf:about="http://www.thetruthaboutcars.com/2005/06/mitsubishi-eclipse-gt/">
```

Figure 3: RDF document for Eclipse

RDF document Validation:

RDF document is validated using the W3C RDF validation service as shown in "Fig. 4".



Validation Results

Your RDF document validated successfully.

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://en.wikipedia.org /wiki/Solar_eclipse	http://www.flickr.com /eclipse#type	"Solar Eclipse"
2	http://en.wikipedia.org /wiki/Solar_eclipse	http://www.flickr.com /eclipse#detail	"moon Between Sun and earth"
3	http://en.wikipedia.org /wiki/Solar_eclipse	http://www.flickr.com /eclipse#total	"moon covers sun"
4	http://en.wikipedia.org /wiki/Solar_eclipse	http://www.flickr.com /eclipse#annular	"moon appears smaller than sun"
5	http://en.wikipedia.org	http://www.flickr.com	"sometimes total, sometimes

Figure 4: W3C RDF Validation service

QUERYING USING SPARQL

Now the RDF document has been queried using SparQl. SparQL, the query language for RDF data [9], is based on graph patterns and sub-graph matching.

The SparQl algebra of Perez and his colleagues [12] defines

1. the unary operators sigma (filtering) and ∏ (projection) that correspond to the SPARQL constructs FILTER and SELECT, respectively, and
2. The binary operators, ∪, ⋈, ⟕ for the SPARQL constructs UNION, AND, and OPTIONAL respectively.

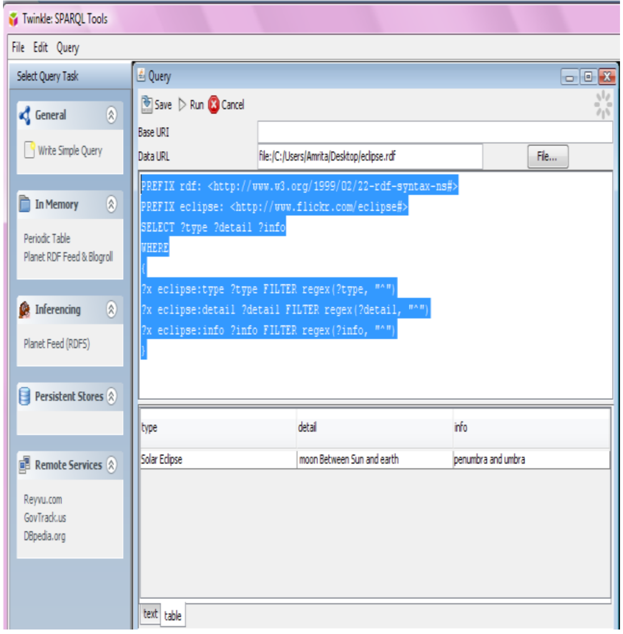
The basic building block from which more complex SPARQL query patterns are constructed is a basic graph pattern (BGP). A BGP is a set of triple patterns which are RDF triples that may contain query variables at the subject, predicate, and object position [10].

A subset of the basic syntax of a SparQL select query is shown below:

BASE < some URI from which relative FROM and PREFIX entries will be offset >
PREFIX prefix_abbreviation: < some_URI >
SELECT some_variable_list
FROM < some_RDF_source_URL >
WHERE {
  { some_triple_pattern .
  another_triple_pattern . }
}

RESULTS

The Results of running Query in Twinkle SparQl Tool are shown in "fig. 5".



Base URI

Data URL file:/C:/Users/Amrita/Desktop/eclipse.rdf

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX eclipse: <http://www.flickr.com/eclipse#>
SELECT ?type ?detail ?info
WHERE
{
?x eclipse:type ?type FILTER regex(?type, "^")
?x eclipse:detail ?detail FILTER regex(?detail, "^")
?x eclipse:info ?info FILTER regex(?info, "^")
}
```

type	detail	info
Solar Eclipse	moon Between Sun and earth	penumbra and umbra

Figure 5: Shows results for eclipse that is of type science and futher categorized as solar

Many other query forms for SparQl were run on this query tool, and the corresponding results were found to be the best optimal. The undesired results were omitted and only the results that were related to the user's query were shown.

CONCLUSION

Since the traditional retrieval techniques are not efficient for homonyms retrieval, this work has focused primarily on this issue and ways to enhance the search results for homonyms have been proposed with the use of RDF, using a self generated repository and one such example studied is of Eclipse. An extension of this through the use of OWL and by additions of more annotations to it, so as to make the retrieval at a more fine level of granularity, is proposed.

The future work will be to develop this using OWL and to achieve intelligent fuzzy retrieval using Fuzzy Logic.

REFERENCES

[1] Jun Zhai and Kaitao Zhou, "Semantic Retrieval for Sports Information Based on Ontology and SPARQL", International Conference of Information Science and Management Engineering, 2010

[2] Davis, Ian, "An Introduction to RDF", http://research.talis.com/2005/rdf-intro/

[3] Michael Grobe, "RDF, Jena, SparQL and the "Semantic Web", http://people.ku.edu/~grobe/SIGUCCS-semantic-web-intro/fp0518-grobe.pdf

[4] K. Selc ̦uk Candan, Huan Liu, and Reshma Suvarna, "Resource Description Framework: Metadata and Its Applications", SIGKDD Explorations, Volume 3, Issue 1 - page 6

[5] Jose K, Marja-Ritta K, Eric Prud H, and Ralph R.," Annotea: An Open RDF Infrastructure for Shared Web Annotations", In Proceedings of WWW 10th International Conference, Hong Kong, May 2001.

[6] McEnery, A. M., Wilson, A., "Corpus Linguistics: An Introduction". Edinburgh University Press, Edinburgh, 2001, Published Online: 14 JAN 2008, Blackwell publications

[7] W.Siberski, J.Z.Pan, and U.Thaden, "Querying the SemanticWeb with Preferences". In ISWC2006

[8] E.Ruckhaus, E.Ruiz,and M.-E.Vidal, "Query Optimization in the Semantic Web". In ALPSWS 2006.

[9] Prud'hommeaux,E.,Seaborne,A, "SPARQL query language for RDF." W3C Recommendation (January2008) Retrieved April 11, 2011, from http://www.w3.org/TR/rdf-sparql-query /

[10] Olaf Hartig, Christian Bizer, and Johann-Christoph Freytag, "Executing SPARQL Queries over the Web of Linked Data", The Semantic Web ISWC 2009 (2009), **Volume:** 5823, Publisher: Springer, Pages: 293–309

[11]Simon Schenk, Steffen Staab, "Networked Graphs: A Declarative Mechanism for SPARQL Rules, SPARQL Views and RDF Data Integration on the Web", WWW 2008 / Refereed Track: Semantic / Data Web - Semantic Web II April 21-25, 2008 · Beijing, China

[12] Pérez, M. Arenas, and C. Gutierrez, "Semantics and Complexity of SPARQL," ACM Trans. Data- base Systems, vol. 34, no. 3, 2009; http://doi.acm.org/10.1145/1567274.1567278.

[13] Berners-Lee T., Hendler J., Lassila, O., 2001, "The Semantic Web", Scientific American, 284: 34–43.

[14] Liyang Yu, "Introduction to the Semantic Web and Semantic Web Services", hapman and Hall/CRC, Taylor & Francis Group publication.

Amrita Bhandari, Author

She is pursuing her ME in Computer Science and Engineering Department, Thapar University, Patiala. She has done her Btech in Computer Science from TIT&S, Bhiwani. Currently she is in ME second year and is doing her ME Thesis work on topic "A Semantic Approach to Efficient Information Retrieval using Annotations".

Mrs. Shalini Batra, Author

Shalini Batra is working as Assistant Professor in Computer Science and Engineering Department, Thapar University, Patiala since 2002. She has done her Post graduation from BITS, Pilani and is persuing Ph.D. from Thapar University in the area of Semantic Web Services and Machine Learning. She has guided nineteen ME s and presently guiding four. She is author/co-author of more than thirty-five publications in national and international conferences and journals. Her areas of interest include Web semantics and machine learning particularly semantic clustering and classification. She is taking courses of Compiler construction, Theory of Computations and Parallel and Distributed Computing.