International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Role of SPARQL in Leveraging Sematic Web Technology

Poornima G. Naik, Kavita S. Oza

Abstract*: Semantic web is not just a matter of translation from HTML to RDF/OWL languages. It is a matter of understanding the content of the web through knowledge graphs. Entities need to be related with relationships. This content is composed of resources (web pages) that contain, for example, text, images and audio. Thus, there is the need of extracting entities from these resources. Currently, most of the web content is in HTML5 format which is a W3C recommendation which enables describing the structure marginally with the help of annotations. The main challenge here is to transform unstructured data from plain HTML files to structured data (e.g RDF or OWL). The current work provides the first hand information for dealing with unstructured heterogeneous data residing on web using Twinkle, a Java tool for SPARQL query execution on FOAF (Friend Of A Friend) document.

Keywords* : Filter, FOAF , Twinkle, RDF, Projection, Ontology, SPARQL.

I. INTRODUCTION

1.1 Current State of Web

The current state of the web is highly unstructured and consists of vast repository of interconnected documents which are presented to end users as a collection of huge inter-linked documents. Extracting a structure from such highly unstructured web poses a big challenge to a researcher. Further, since the content is available for public access, quality of the content posted on WWW cannot be validated and guaranteed to be reliable. Also, the persistence of documents cannot be uniformly guaranteed. HTML's simplicity comes at a cost of interoperability which implies HTML documents are human readable but extensive ground work is desirable to make them machine readable and inter-operable by different software's. This is how XML emerged adding structuredness to unstructured HTML data in the form of DTD and Schema. The current state of the web is mature enough owing to the new technologies such as XML, Ontology, SPARQL etc. to name a few which strive to ingest some sort of structuredness and semantics to the otherwise unstructured and heterogeneous web.

Revised Manuscript Received on February 15, 2020.
* Correspondence Author
Poornima G. Naik, Professor in the Department of Computer Studies, CSIBER, Kolhapur.
Kavita S. Oza, assistant professor at Department of Computer Science, Shivaji University, Kolhapur

1.2 Introduction to SPARQL

SPARQL plays a key role in executing queries against heterogeneous data sources employing its native RDF format or which is transformed into RDF format by some middleware application. SPARQL operates on RDF graphs and mainly employs the logical operations conjunction and disjunction for unleashing the unknown relationships between the data and generates the results which can be result sets or themselves be RDF graphs.

SPARQL is the query language of the Semantic web which enables

- Pulling the data from both structured as well as semi-structured data
- Data exploration by querying unknown hidden relationships between data
- Performing complex joins of heterogeneous databases employing a simple query
- Transforming RDF data from one vocabulary to another

The following section describes Twinkle, a Java-based tool for the execution of SPARQL queries.

1.3 Working of Spark query generator and Executor Model.

Model is developed for execution of Spark query which accepts FOAF document, generates Spark query on fly and employs Twinkle, a Java based tool for execution of Spark queries. Figure 1 depicts working of SPARQ Query Generator and Executor model.

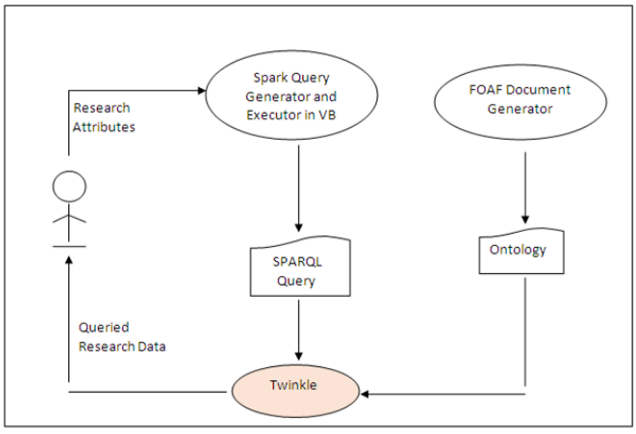


Figure 1 Working of SPARQ Query Generator and Executor module

✓ Depending on the subject area selected by an end user HTML document generator initially searches the local file system for the required FOAF file.

The naming convention adopted for FOAF document is <subject name>FOAF.xml which is in XML file format.

✓ If the file exists then it is used otherwise FOAF document is dynamically generated by invoking HTML Page Extractor and Multi-threaded Web Crawler module.

✓ The foaf document created in the above step is input to spark query generator tool which generates dynamically a spark query.

The spark query is executed by employing a java based tool twinkle which generates the required output.

II. RESULTS AND DISCUSSIONS

2.1 Generating a Spark query based on end user input.

A GUI has been designed for the purpose in VB6 for accepting the research journal specific attributes from an end user. Currently, the user can query the research data pertaining to the following attributes:

- Journal type which can be either national or international
- Information about the journal corresponding to one or more of the following fields:
  - Title
  - Volume
  - Issue
  - Charges
  - e-ISSN
  - p-ISSN
- UGC Recommended Journals
- Maximum impact factor in the available journals
- Impact factor in the given range.
- Minimum processing charges
- Processing charges in the given range.

In each case the user can view the generated Spark query.

2.2 Executing the generated Spark query using Twinkle

TWINKLE is the most popular tool used to execute SPARQL Query. To use the tool it requires jdk1.5 or higher to be installed on the system. Figure 2 shows command-line for the execution of Twinkle tool.

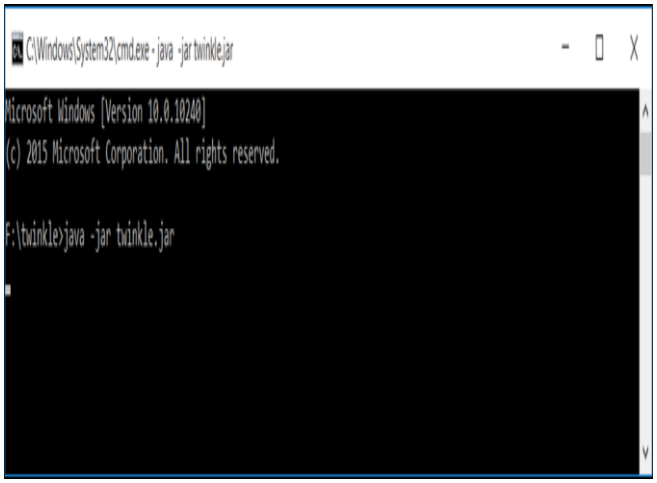


Figure 2 Execution of Twinkle Tool Through Command-Line

Most forms of SPARQL query contain a set of triple patterns called a basic graph pattern. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable. A basic graph pattern matches a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is RDF graph equivalent to the subgraph. The SPARQL query along with the FOAF document generated is input to a java tool, Twinkle which fires SPARQL query on FOAF document to generate a desired output to an end user. Figure 3 depicts the "Spark Query" main menu structure.

Figure 3. Structure of 'Spark Query' Menu of Semantic Web Application

Figure 4 depicts the GUI for SPARQL Query Generator.

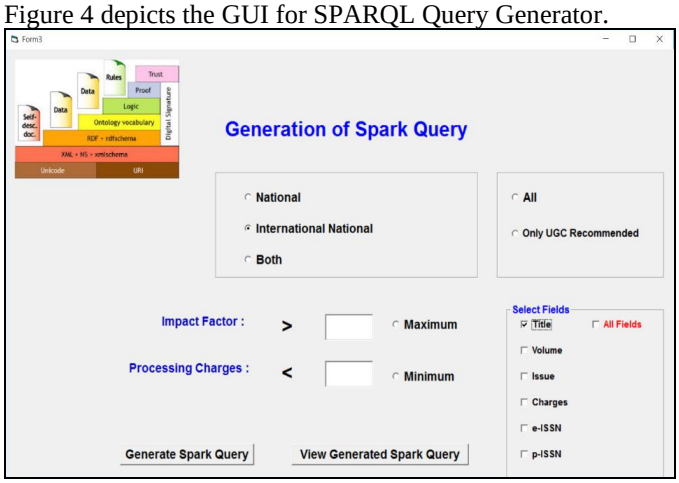
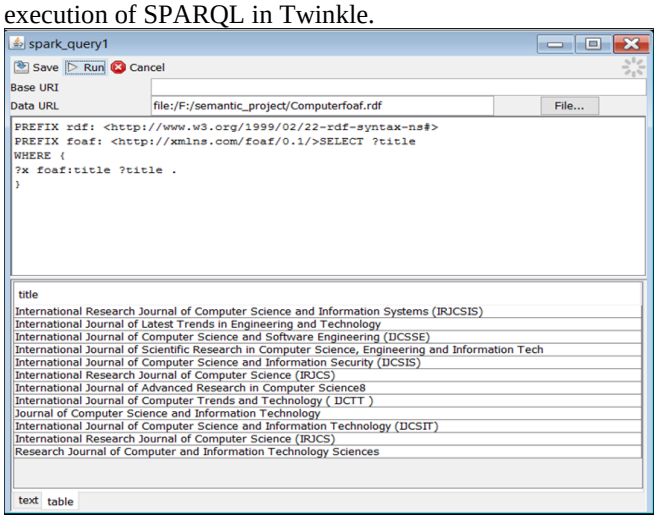


Figure 4. SPARQL Query Generator

The following section highlights execution of few sample SPRQL queries employing projection, selection and rewriting rule (employing Filter variable). The generated SPARQL queries are further executed by Twinkle, java tool for executing SPARQL queries.Figure 5(a)-5(b) depict the execution of SPARQL in Twinkle.



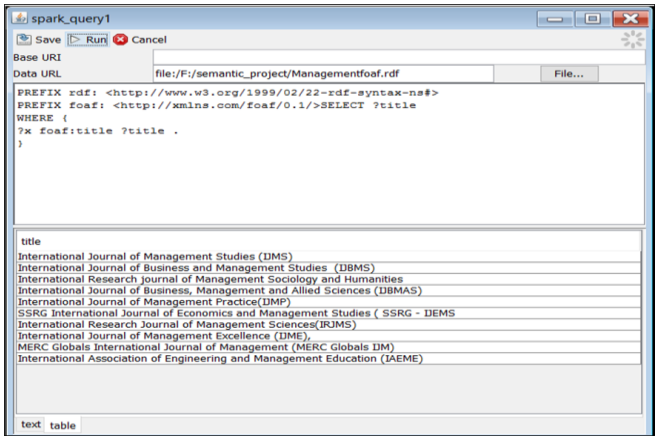


Figure 5(a)-5(b) Execution of SPARQL for Projection in Twinkle in Table Format for Research Journals in Computer and Management.

Applying Projection

Query 1: The following query will find the subject, predicate (properties) and object of the research journals.

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT * WHERE {?s  ?p  ?o}
```

Figure 6. depicts the execution of SPARQL in Twinkle.

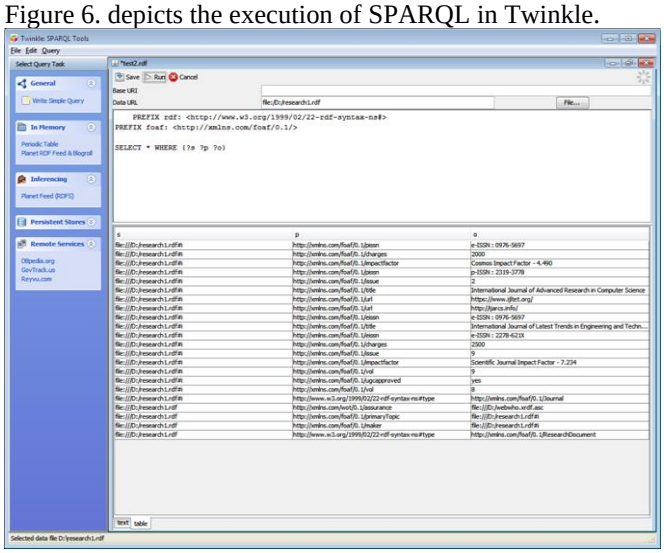


Figure 6. Execution of SPARQL in Twinkle for Generating RDF Triplet

Query 2: The following query will find the title, volume, issue number, charges, UGC approved status, impact factor, eissn, pissn and URL of the research papers in various journals.

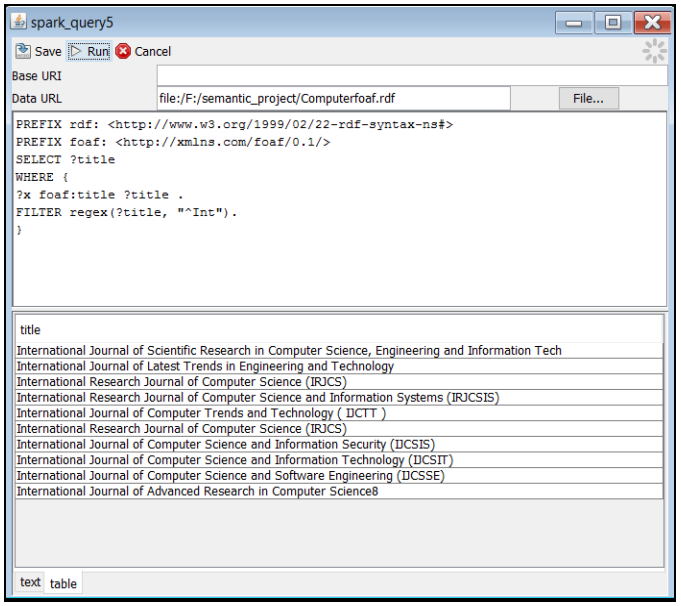
```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?title ?vol ?issue ?charges ?ugcapproved
?impactfactor ?eissn ?pissn ?url
WHERE {
  ?x foaf:title ?title .
  ?x foaf:vol ?vol .
  ?x foaf:issue ?issue.
  ?x foaf:charges ?charges.
  ?x foaf:ugcapproved ?ugcapproved.
  ?x foaf:impactfactor ?impactfactor.
  ?x foaf:eissn ?eissn.
  ?x foaf:pissn ?pissn.
  ?x foaf:url ?url.
}
```

Query 3: The following query will find the pattern matching using regular expressions for filtering research papers in international journals.

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?title
WHERE {
  ?x foaf:title ?title .
  FILTER regex(?title, "^Int").
}
```

Figure 7. depicts the execution of SPARQL in Twinkle for pattern matching.



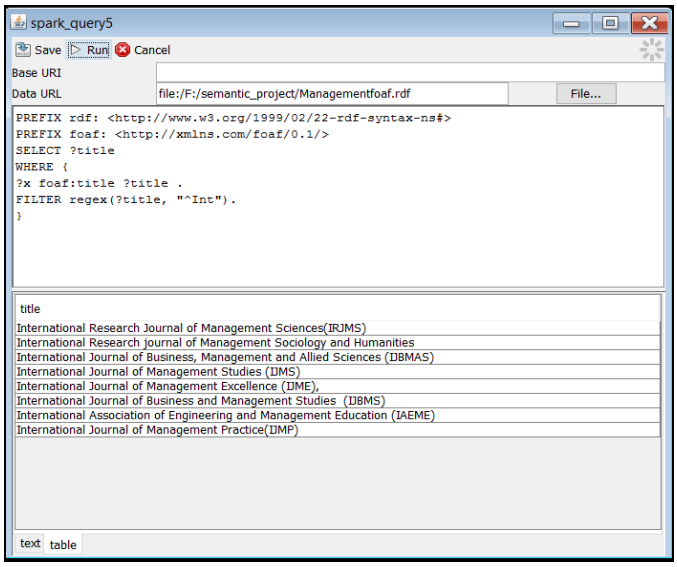


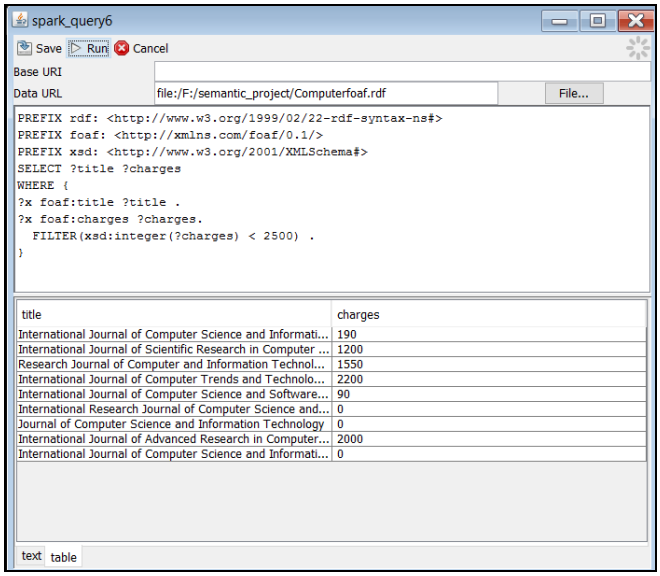
Figure 7. Execution of SPARQL in Twinkle for Pattern Matching

Query 4: The following query shows the method for converting string to integer or for employing an expression in a numeric for finding research papers in various journals with charges less than 3000.

CONVERTING STRING TO INTEGER

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?charges
WHERE {
  ?x foaf:charges ?charges .
FILTER(xsd:integer(?charges) < 2500) .
}
```

The execution of the query in Twinkle is shown in Figure 8.



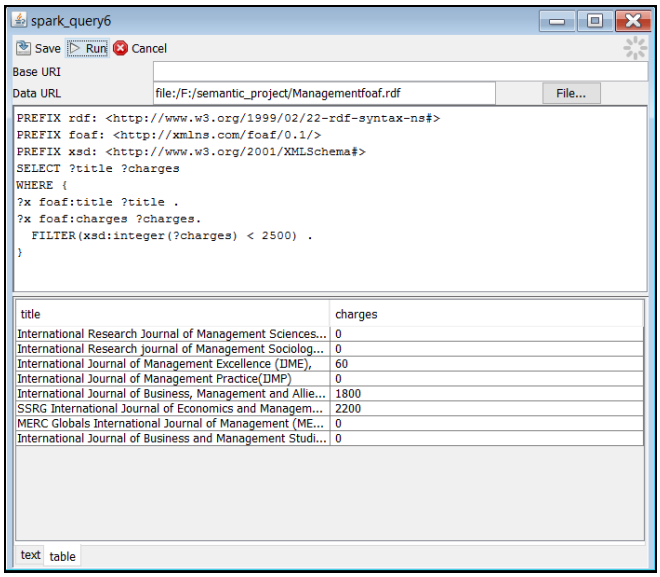


Figure 8. Execution of Filter SPARQL in Twinkle for Filtering

The logic for finding the research journals with highest impact factor is selecting the first item after sorting the impact factors in descending order.

FINDING MAXIMUM CHARGE

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?impactfactor
WHERE { ?x foaf:impactfactor ?impactfactor }
ORDER BY DESC(xsd:integer(?impactfactor)) LIMIT 1
```

Similarly, for finding research journals with lowest charges, select the first item after sorting the charges in ascending order.

FINDING MINIMUM CHARGE

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?charges
WHERE { ?x foaf:charges ?charges }
ORDER BY ASC(xsd:integer(?charges)) LIMIT 1
```

III. CONCLUSION AND SCOPE FOR FUTURE WORK

The current research provides the first hand information for dealing with unstructured heterogeneous data residing on web with an emphasis to Twinkle, a Java tool for SPARQL query execution on FOAF document. The research can be extended further to retrieve the text from the images employing OCR tools. Also, image scraper or web scraper can be adopted for extracting large amounts of information from the website which involves downloading several web pages or the entire website which may include text from pages or HTML or both HTML and images. Some of the best web scraper tools are import.io, webhose.io, scrapehub, parsehub, visualscraper, spinn3r etc.

REFERENCES

1. Matthias Palmér "Learning Applications based on Semantic Web Technologies." Doctoral thesis, Stockholm, Sweden 2012. Available at: https://www.diva-portal.org/smash/get/diva2:564709/FULLTEXT01.pdf
2. Marcelo Arenas,Jorge Pérez "Querying Semantic Web Data with SPARQL." Department of Computer Science, Universidad de Chile, Available at:
3. http://www.csd.uoc.gr/~hy561/papers/formalization/Querying%20Semantic%20Web%20Data%20with%20SPARQL.pdf
4. Prasad Kulkarni "Distributed SPARQL query engine using MapReduce." University of Edinburgh , 2010. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.186.2063&rep=rep1&type=pdf
5. Mulugeta Mammo "Distributed SPARQL over Big RDF Data, A
6. Comparative Analysis Using Presto and MapReduce." Arizona State
7. University, December 2014, Available at:
8. https://repository.asu.edu/attachments/143388/content/Mammo_asu_0010N_14524.pdf

AUTHORS PROFILE

Dr. Poornima G. Naik, received her M.Sc. degree in Physics and Mathematics and Ph.D. degree in physics from Karnataka University, Dharwad. She received MCA degree from IGNOU with first class Distinction. Currently, she is working as Professor in the Department of Computer Studies, CSIBER, Kolhapur. Her areas of interest are network security, soft computing and cloud computing. She has participated in several national and international conferences, authored 20+ books on various cutting edge technologies in IT and has published more than 70 papers in International and national journals of repute. She is a prolific technical writer with excellent communication, analytical and technical skills. She is a recipient of prestigious Dr. APJ Abdul Kalam Life Time Achievement National Award for remarkable achievements in the field of Teaching, Research & Publications awarded by International Institute for Social and Economic Reforms, Bangalore.

Dr. Kavita S. Oza has received her PhD from Shivaji University in Computer Science. Currently she is working as assistant professor at Department of Computer Science, Shivaji University, Kolhapur. She is life member of CSI. Her research interest is machine learning, algorithms and Text mining. She has more 30 research publications to her credit. Two students have been awarded PhD under her guidance and four are pursuing the same.