

# SPARQL: SEMANTIC INFORMATION RETRIEVAL BY EMBEDDING PREPOSITIONS

Sneha Kumari<sup>1</sup> Dr.Rajiv Pandey<sup>2</sup> Amit Singh<sup>3</sup> Himanshu Pathak<sup>4</sup>

<sup>1</sup>Scholar, Amity University, Lucknow, India

<sup>2</sup>Asst. Professor, Amity University, Lucknow, India

<sup>3</sup>Scholar, Amity University, Lucknow, India

<sup>4</sup>Scholar, Amity University, Lucknow, India

## ABSTRACT

*Semantic web document representation is formulated using RDF/OWL. RDF representation is in the form of triples & OWL in form of ontologies. The above representation leads to a data set which needs to be queried using software agents, machines. W3C has recommended SPARQL to be the de facto query language for RDF. This paper proposes to suggest a model to enable SPARQL to make search efficient, easier and produce meaningful search distinguished on the basis of preposition. An RDF data source primarily consist of data that is represented in the form of a Triple pattern which has an appropriate RDF syntax and further results into an RDF Graph .The RDF repository stores the Data on Subject, Predicate, Object Model. The Predicate may also be thought of as a property linking the Subject and Object. This paper shall evaluate the information retrieval by incorporating preposition as property.*

## KEYWORDS

SPARQL, N-Triples, OWL, RDF.

## 1. INTRODUCTION

“SPARQL is a query language for RDF data on the semantic web with formally defined meaning” [1].

SPARQL a query language for RDF is used to return and manipulate data from databases that are stored in Resource Description Format. It is the recommendation of (DAWG) Data Access Working Group on RDF under World Wide Web Consortium, and is recognized as one of the key technologies of the semantic web. [1]

Almost all RDF data repositories support SPARQL either directly or through dedicated SPARQL tools. The main feature of computing query results through SPARQL is achieved by sub-graph matching. These sub-graph are the generation based on RDF subject predicate object definition. RDF graphs are interpreted as nodes and edges.

A lot has been researched on SPARQL and its query support to RDF repositories however it is of concern that no headway has been achieved in the semantic search based on preposition.

This paper therefore proposes to incorporate prepositions in the SPARQL search pattern. This paper is divided into five sections i.e.

- (1) Introduction
- (2) PL/SQL approach to preposition based search
- (3) SPARQL querying the RDF repository
- (4) Incorporating prepositions in triple patterns
- (5) Conclusion.

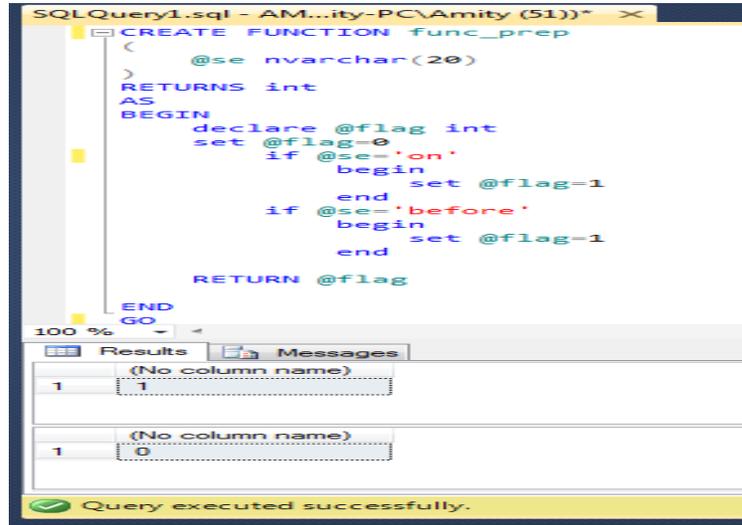
## 2. PL/SQL APPROACH TO PREPOSITION BASED SEARCH

The below described approach is based on PL/SQL whereby that we have developed a function prep by CREATE FUNCTION func\_prep. This function takes an argument which is the preposition "on" from the search pattern "Books on Steganography" the two keywords (Books, Steganography) can be interpreted by the search engine thru metadata definition but it is important to identify the existence of the preposition "on". The PL/SQL function below can be passed, any preposition like(in , on , after , before , ago , since , at ,till , until ,towards and etc.),.The function then returns the flag value as '1' for the existence of preposition and '0' for non existence. The same is demonstrated thru the Fig 1, 2 and 3.

```
CREATE FUNCTION func_prep
(
    @se nvarchar(20)
)
RETURNS int
AS
BEGIN
    declare @flag int
    set @flag=0
    if @se='on'
        begin
            set @flag=1
        end
    if @se='before'
        begin
            set @flag=1
        end

    RETURN @flag
END
```

GO

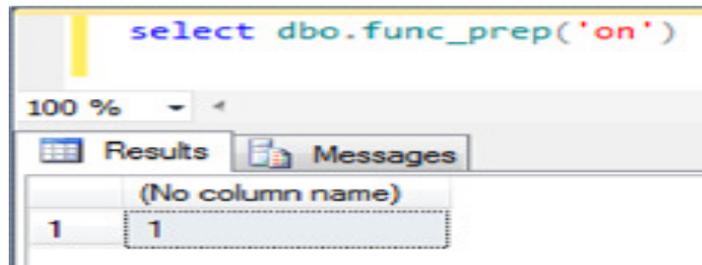


```
CREATE FUNCTION func_prep
(
    @se nvarchar(20)
)
RETURNS int
AS
BEGIN
    declare @flag int
    set @flag=0
    if @se='on'
    begin
        set @flag=1
    end
    if @se='before'
    begin
        set @flag=1
    end
    RETURN @flag
END
GO
```

(No column name)	
1	1
(No column name)	
1	0

Query executed successfully.

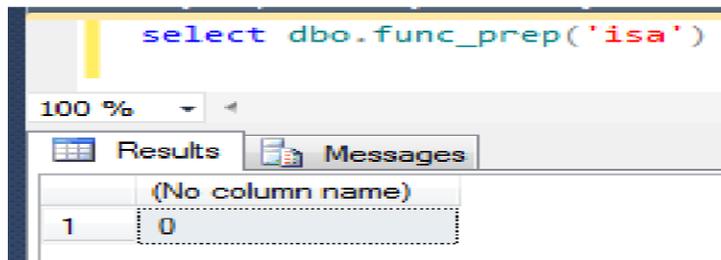
Figure 1. Screen shot of SQL func\_prep function. Here preposition found then output is 1.



```
select dbo.func_prep('on')
```

(No column name)	
1	1

Figure 2. Passing a preposition to SQL function. Here, preposition not found and output is 0.



```
select dbo.func_prep('isa')
```

(No column name)	
1	0

Figure 3. Validation result return as True or False.

The preposition existence is only highlighted by the above PL/SQL snippet but the semantic matching still remains to be undertaken by a semantic tool not being the part of PL/SQL. This approach exhibits a major limitation thus we need to look up to a SPARQL based search architecture which can return results based on preposition. The same is explored in the subsequent sections.

### 3. SPARQL QUERYING THE RDF REPOSITORY

SPARQL the query language for the RDF Semantic data sources permits the under mentioned and many more –

- It retrieves data from Subject Predicate Object represented RDF data.
- Explore data by querying unknown relationships.
- Enables join operations on disparate databases.
- It has aggregate functions like count() , aggregate() etc.

It is observed that SPARQL exhibits a major limitations when it comes to searches based on prepositions .The Fig(4) describes SPARQL search architecture and it is observed that there is no provision to distinguish between the RDF graphs on the basis of prepositions

#### 3.1. Basic Structure of SPARQL query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>
SELECT distinct ?Event1_location
WHERE
{
    ?Presentation1 rdf:type swrc:Presentation .
    ?Presentation1 swrc:isPresentedAt ?Event1 .
    ?Event1 rdf:type swrc:Event .
    ?Event1 swrc:location ?Event1_location .
    FILTER REGEX (str(?Event1_location), 'IIT Kanpur', 'i')
}
```

The Query above clearly signifies that each query embeds a Subject Predicate and Object component. The components i.e. Presentation isPresentedAt IIT Kanpur being the Subject Predicate and Object respectively.

#### 3.2. RDF dataset explored:

An RDF data source primarily consist of data that is represented in the form of a Triple pattern which has an appropriate RDF syntax and further results in to and RDF Graph .each of theses are explained in the following paragraphs.

Triple Patterns [2] [3]: A triplestore is a domain specific database for the storage and retrieval of triples, a triple being a data entity composed of subject-predicate-object. Fig (4) is used to show the subject – predicate – object structure of RDF in Triple pattern data store. The tool used for generating the RDF data model is RDF validator of W3C [4]

RDF syntax of subject-predicate-object data model: the RDF syntax Fig (5) distinguishes to state that `http://www.facultydesc.demo/AMITY/Computerscience` is subject, Subject can be namespaces of XML or URI as `http://www.facultydesc.demo/AMITY/Computerscience`, Predicate is the property 'name' and 'phone' of the faculty represented thru URI `http://www.facultydesc.demo/AMITY#name,URI http://www.facultydesc.demo/AMITY#phone` respectively.

Triples of the Data Model			Feedback
			Back to Validator Input
Number	Subject	Predicate	Object
1	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#name</code>	"Rajiv Pandey"
2	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#phone</code>	"94673526"
3	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#Addresshno</code>	"S1-141"
4	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#AddressStreet</code>	"Suraksha-1"
5	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#Addresscolony</code>	"Eldeco Udyan-2"
6	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#AddressCity</code>	"Lucknow"
7	<code>http://www.facultydesc.demo/AMITY/Computerscience</code>	<code>http://www.facultydesc.demo/AMITY#Qualification</code>	"M.Tech."

Figure 4. subject-predicate-object representation of a RDF data model.

```

The original RDF/XML document
1: <?xml version="1.0"?>
2: <rdf:RDF
3: xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4: xmlns:faculty="http://www.facultydesc.demo/AMITY#">
5: <rdf:Description rdf:about="http://www.facultydesc.demo/AMITY/Computerscience">
6: <faculty:name>Rajiv Pandey</faculty:name>
7: <faculty:phone>94673526</faculty:phone>
8: <faculty:Addresshno>S1-141</faculty:Addresshno>
9: <faculty:AddressStreet>Suraksha-1</faculty:AddressStreet>
10: <faculty:Addresscolony>Eldeco Udyan-2</faculty:Addresscolony>
11: <faculty:AddressCity>Lucknow</faculty:AddressCity>
12: <faculty:Qualification>M.Tech.</faculty:Qualification>
13: </rdf:Description>
14: </rdf:RDF>
    
```

Figure 5 : RDF/XML syntax based on tripple pattern.

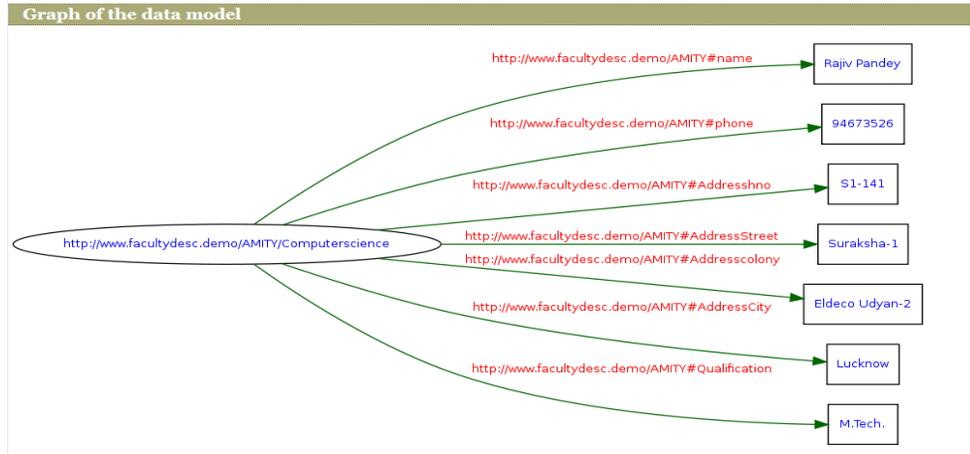


Figure: 6 RDF graph

Graph of RDF triple: The resultant graph of the RDF triples is represented in Figure: (6) which shows the nodes and edges.

### 3.3. INCORPORATING PREPOSITIONS IN TRIPLE PATTERNS

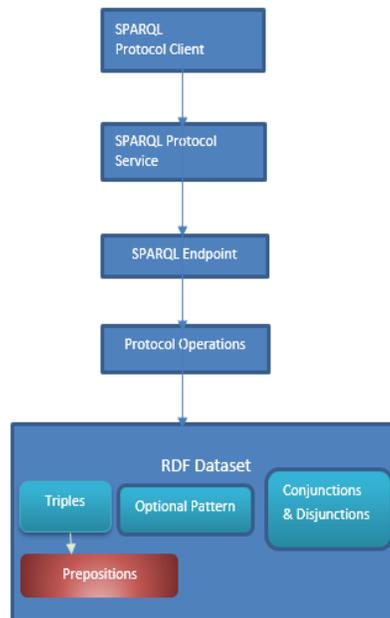


Figure 7. Architectural representation of a SPARQL search.

#### 3.3.1 Architectural View [6] [7]

- SPARQL Protocol client: An HTTP client which is capable to send HTTP request to SPARQL Operations.
- SPARQL Protocol service: This service is more of an application or server which can accept HTTP request convert it into appropriate SPARQL request & when SPARQL operation generates the response the SPARQL response shall be approximate returned as HTTP response to the HTTP client.
- SPARQL Endpoint: “The URI at which a SPARQL Protocol Service listen for request “[8]. This may be understood as port which it is listening to the HTTP request.
- SPARQL Protocol operation: The SPARQL protocol operation consist of various methods which can accept the request .This process it to query the underlying RDF dataset & also capable to generate response .This operation protocol is built over HTTP .Therefore it supports the GET & POST method of HTTP.
- RDF Dataset: The RDF dataset store represents information as graphs, which are resultants of Triples. Triples are the description based on Subject, Predicate & Object Paradigm.

### 3.3.2. Preposition as property in SPARQL:

We are using the tool SPARQL query builder [5] to generate the SPARQL query based on the property which serves as a predicate.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX swrc: <http://swrc.ontoware.org/ontology#>
SELECT distinct ?Event1_location
WHERE
{
    ?Presentation1 rdf:type swrc:Presentation .
    ?Presentation1 swrc:isPresentedAt ?Event1 .
    ?Event1 rdf:type swrc:Event .
    ?Event1 swrc:location ?Event1_location .
    FILTER REGEX (str(?Event1_location), 'IIT Kanpur', 'i')
}
```

In the above SPARQL query the property “isPresentedAt” has been considered for the purpose of demonstration, we have to add prepositions as properties which will result in Semantic search. The Semantic search based prepositions is only possible when the data source specifies the dataset description based on prepositions. This class will help to check the preposition’s in between the words and generate the result.

Any data repository which has a schema to cater for preposition related description can generate result based on the preposition property. In case of preposition, the predicate described as property in the triple store will also be matched for the preposition in between the words.

The SPARQL query was generated by VIVO SPARQL online query builder.

The screenshot shows the VIVO SPARQL Query Builder interface. It features a table with three columns: Subject, Predicate, and Object. The first row contains 'swrc:Presentation' in the Subject column, 'swrc:isPresentedAt' in the Predicate column, and 'swrc:Event' in the Object column. The second row contains 'swrc:Event' in the Subject column, 'swrc:location' in the Predicate column, and 'IIT Kanpur' in the Object column. Below the table is a 'Generate Query' button. The resulting SPARQL query is displayed below the button, matching the code provided in the text above.

Figure 8. VIVO SPARQL builder demonstrating the predicate as property.

When the triple data store will contain the triple “Books on Steganography” using the predicate “on” stored as property. Then the SPARQL query will return a more meaningful result which will be based on the preposition.

## 4. CONCLUSION

This paper has primarily described how RDF data store need to be represented as a schema in which the prepositions will be used as properties. The properties will be then interpreted by the SPARQL engines. The property based schema can extend to OWL ontologies which are RDF serialized.

The SPARQL engine will however only return search based on preposition when the underlying data set is got the appropriate schema. Thus all data sets of the future will need to be RDF repositories. It is therefore a subject of research to have software agents which can convert Non-RDF datasets to RDF data sets. The agents can a part of the SPARQL engine itself which can interpret Non-RDF datasets to be RDF data sets only for the purpose of query.

## REFERENCES

- [1] [www.w3.org/TR/2009/WD-sparql-features-20090702/](http://www.w3.org/TR/2009/WD-sparql-features-20090702/)
- [2] <http://en.wikipedia.org/wiki/Triplestore>
- [3] <http://apachebite.com/2012/04/>
- [4] <http://www.w3.org/RDF/Validator/>
- [5] <http://vivo-onto.slis.indiana.edu/SPARQL/>
- [6] <http://www.w3.org/TR/sparql11-protocol/#rfc2616>
- [7] <http://www.w3.org/TR/sparql11-query/#rdfDataset>
- [8] <http://www.w3.org/TR/2012/WD-sparql11-protocol-20120105/>
- [9] STEFAN DECKER et.al. Framework for the Semantic Web: An RDF Tutorial, Stanford University.
- [10] Marko A Rodriguez, an RDF/OWL Tutorial, Digital Library Research and Prototyping Team Alamos National Laboratory, Los Alamos, New Mexico, June 2007.

### Authors

Sneha Kumari is pursuing M.C.A from Amity University Lucknow campus in Amity Institute of Information Technology. She will strive to write more research papers on Semantic Web. She is a Microsoft Certified Professional Developer in web technology.



Dr. Rajiv Pandey is a Senior faculty at Amity University Lucknow campus in Amity Institute of Information Technology. His research area is Semantic Web technologies and ontology. He has published large number of research papers in various International and National Journals.



Amit Singh is pursuing M.C.A from Amity Institute of Information Technology, Amity University Lucknow. It is his first paper and he will strive to write more research papers on Semantic Web related to Provenance. He is a Microsoft Certified Web Developer in various platform



Himanshu Pathak is pursuing M.C.A from Amity University Lucknow campus in Amity Institute of Information Technology .He will strive to write more research papers on Semantic Web related to Provenance. He has published research papers in various International Journals

