ISSN No:-2456-2165

Efficient Semantic Web Data Searching Using Virtual Documents Algorithm

Nay Nandar Linn[1], Thinn Thinn Win[2]
[1] Information Technologies Support and Maintenance
University of Computer Studies, Taungoo, Myanmar

Abstract:- Semantic Web technologies have now been applied to build new or semantic enhanced real-world applications. Semi-structured information which integrates metadata is becoming accessible on the Internet. Current search methods do not tackle the level of semantic matching that is needed to align user information with vocabulary elements and introduce a barrier to the creation of Linked Information on the Web. To this end, we describe a semantic search method in which keywords can be adapted to query semantic web data. This approach converts keyword queries automatically into formal structured queries such that end users can use keywords to perform semantic searches. In particular, this paper focuses on constructing and querying the semantic web data in a virtual document. Finally, we provide both keyword-based and more sophisticated search interfaces to retrieve the relevant data objects.

Keywords:- *SPARQL, RDF Graph, Virtual Document, Semantic Search.*

I. INTRODUCTION

The enormous amount of accessible information in the World Wide Web caused excessive demand for tools and techniques capable of handling semantic data. The current information retrieval practice relies mostly on keyword-based searching over full text data. Such a model, however, lacks the actual semantic information in the text. The Resource Description Framework (RDF) and ontologies are described for knowledge representation, which are the background of semantic web applications. Such metadata can benefit from both the extraction of information and the retrieval processes. Several query languages are planned for semantic querying. SPARQL is actually the state of the art Semantic Web query language.

Structured and semi-structured data sources are accessible and are being published on the Web. Semantic annotations for heterogeneous tools on the Internet are displayed in RDF, and RDF spaces are searched using RDF query language, called Simple Protocol and SPARQL Query Language (SPARQL). All legacy data as well as new data were made available in triple format RDF. This representation is worthwhile and feasible to establish mappings between RDF data originating from various legacy sources, resulting in very broad RDF repositories.

Within RDF triple representations, the vast amount of highly structured data is available. In particular, the RDF (Resource Description Format) Semantic-Web data model is gaining popularity for applications on scientific data such as biological networks [17], social Web 2.0 applications[11], large-scale knowledge bases such as DBpedia [17] or YAGO [6], and more generally, as a light-weight representation for the "Web of data" [1].

A list of RDF data consists of a series of the subject-property-object triples, in short, SPO triples. In the terminology of Entity Relationship, a triple SPO refers to a pair of entities linked by a named relation or to an entity linked to the value of a named attribute. Many of the largest sets of RDFs include over one billion triples [6].

At first, they face inconsistencies, incompleteness and redundancies for the sake of different methodologies used in RDF graph generation. These are partly addressed by the quality assessment approach, such as through provenance tracking. Secondly, while data quality will be good, the discovery and exploring of RDF graphs requires familiarity with the structure and knowledge of the exact URIs.

In the Information Retrieval System, a considerable amount of work was performed on finding and extracting RDF triples. Current user support approaches in the search process do not tackle the degree of semantic matching needed to search concepts on the Web. This system addresses innovative methods of information retrieval (IR) that can add a new dimension to the task of searching large RDF graphs. This paper suggests an approach focused on the methods of collecting information which can add a new aspect of discovering large RDF graphs.

II. RELATED WORK

Falcon Search [7] provides an ontology search engine based on keyword, which recovers ontology concepts with a combination of term-based significance and ratings. Falcon indexes all terms from virtual document to its classes and properties for each entity, and combines them with the inverted index from terms to concepts. The term-based similarity determines the ranking scores between those entities-related virtual documents and the user keyword query. Moreover, [15] it based on Falcon's design search engine in this multithreaded architecture crawler dereferences URI with content negotiation and downloading of RDF documents, parse by machine these

documents. It provides user search information via the UI. First, clients enter his keyword query on the web page. All information like RDF triples, is retrieved from the semantic web and the document URI is stored in quadruple store.

Swoogle [18] is a crawler-based RDF document-discovery, indexing, and query framework. Swoogle primarily provides a search for Semantic Web documents and words (i.e., class and property URIs). The index is built against RDF files by extracting the keywords from URIs. It allows users to define queries containing document-level metadata conditions, and also allows users to search for Semantic Web.

Sindice[10] is an entity-centered search and query tool for the Linked Data Network that ranks entities by the occurrence of keywords associated with the entities present in the dataset. Sindice gathers RDF documents, and indexes the resources found on URIs and keywords of information. This index allows different URIs to search resources which actually find the same thing in the real world. It indexes keywords and URIs against RDF files, and uses inverted index scheme to contain certain keywords and URIs.

A graph-based query language is supported by the NAGA semantic search engine [8] to query the underlying knowledge base represented as a list. The KB is automatically developed by a tool that extracts information from Web sources.

SPARK[16] is a retrieval system that can use SPARQL queries created from keywords to query semantic data. It consists of three main steps: term mapping, constructing query graphs and ranking queries. The term mapping phase attempts to match query conditions to ontological resources. In the query construction step, the user query is evaluated and the words are related in order to obtain SPARQL queries with the missing connections. The queries are then evaluated with a probabilistic ranking model, such that higher levels are obtained for the more likely SPARQL queries.

One of the biggest challenges in IR for RDF data lies in evaluating the relatedness between an entity and the users's intent. It relies on term frequency weighting functions based on the assumption. The more frequently a term occurs, the more related it is to the topic of the documents[3].

This paper[13] proposes a semantic search system based on keywords that uses a description graph structure to explore RDF data and provides relevant results. It's graph structure is constructed automatically over RDF data for efficient graph explorations.

III. BACKGROUND

A. RDF Data Model and SPARQL

Let I, B, L, and V denote pair wise disjoint infinite sets of Internationalized Resource Identifiers (IRIs), blank nodes, literals, and variables, respectively. Let IB, IL, IV, IBL, and IVL denote I ∪ B, I ∪ L, I ∪ V, I ∪ B ∪ L, and I ∪ V ∪ L, respectively. Set IBL elements are also termed RDF words. The notions are as follows: RDF triple, RDF graph, triple pattern, graph pattern, and SPARQL query. A sample RDF graph used for the following examples is shown in Fig 1.

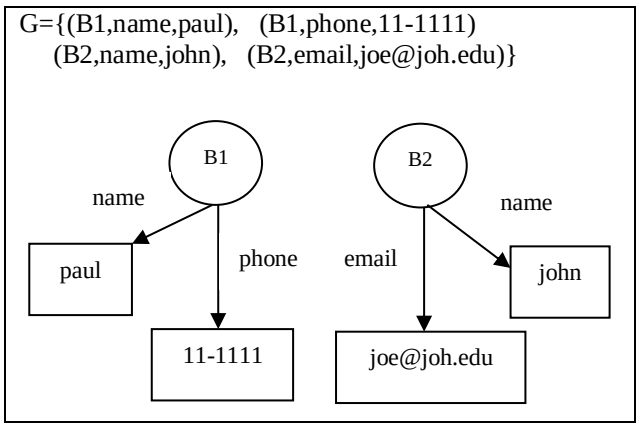

Fig 1:- Sample RDF Graph

B. Semantic Association

Semantic associations are dynamic relationships between individuals involved in the resources. Some intermediate entities and relationships are involved in most important semantic associations. It helps the user interact with various people, locations and events. We provide definitions for the formalization of semantic association adapted from [12] to define the semantic association.

IV. PROPOSED SYSTEM DESIGN FOR SEMANTIC SEARCH

In this model, First of all, the system is accepted keyword queries related to computer science research fields domain from the user, and then translate the SPARQL query language compatible with keyword query. In order to search efficiently and effectively from a given keywords query, the virtual documents from RDF graph nodes are constructed and stored them into the knowledge base. The SPARQL query is carried out against the knowledge base (KB), which returns a list of instance tuples satisfying the query.

This system consists of three components in this semantic search: model creation, semantic analysis and semantic search. Our approach to semantic information retrieval can be seen as an evolution of the traditional keyword-based retrieval techniques, where a semantic knowledge base replaces the keyword-based index. Fig 2 demonstrates the overall architecture of searching semantic Web data over RDF Graph.

ISSN No:-2456-2165

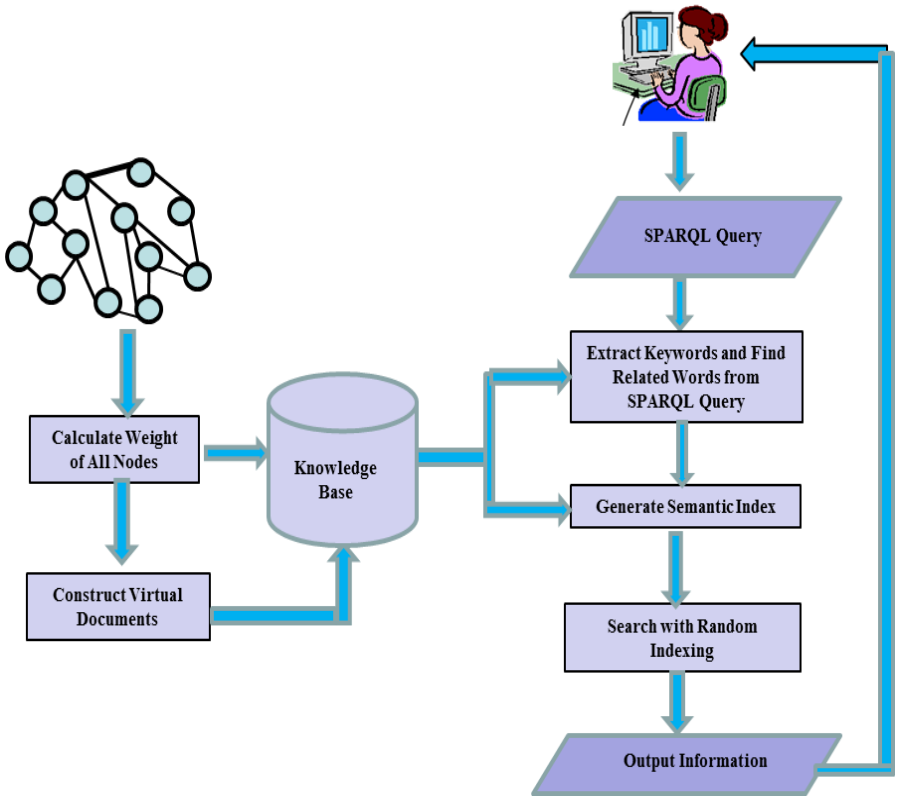

Fig 2:- System Design of Semantic Searching

V. SEMANTIC ANALYSIS

In this semantic analysis, a document is known as a set of words derived or inferred directly from the content of the document. These weighted terms treated as the vector dimensions, reflect their significance to the document. In a preprocessing step, the words involved should first be normalized. The virtual document is rendered with a series of weighted terms, in which rational numbers are weights. A literal node definition is a set of terms which are derived from its lexical form. The terms are derived for a named entity from the local URI language.

When a document contains a term, a non-zero weight is assigned for that term to its value in the vector. Some of the most common methods for measuring this weight is *term frequency-inverse document*. Weights increase proportionally to the number of the term appearances in the document, in this weighting scheme, but are scaled down by the frequency of the term in the RDF graph nodes.

To perform querying, vectors assigned to documents can be compared to query vectors constructed from search terms. They can also be compared against other documents' vectors to determine the similarity between documents. This approach is used in Computer Science Research Field's Related Articles. Thus standard vector-based search is based only on the existence and frequency of terms in documents and does not find any additional terms-related information on itself. Terms from the title and abstracts that extract RDF Data Graph are calculated term frequency, inverse document frequency, weighting and maximum term frequency normalization.

A. Term Frequency

Let D = {d_1, . . . , d_n} be a set of documents and T = {t_1, . . . , t_m} set of different words contained in D. A document is then interpreted as a m-dimensional vector. Let TF (d, t) denote the frequency of term $t \in T$ in document $d \in D$.Then the vector representation of a document d is

$$TF(d,t) = 1 + \log(1 + \log(freq(d,t))) \quad (1)$$

Each word in a document is assigned a weight for that word, which depends on the number of the term occurrences in the document. A ranking, based on the weight of t in d, can be determined between a query term t and a document d. The easiest method is to assign the weight in document d to be equal to the number of occurrences of term t.

Terms are simply words. First of all, we removed stop words. Term frequency in retrieval models is the most significant retrieval signal. The linear scaling at term frequency is generally accepted as putting too much weight on repeated occurrences of a term.

B. Inverse-Document Frequency

The inverse frequency of a document (IDF) is a statistical weight used to measure the importance of a word in a set of text documents. A term's document frequency DF is determined by the number of documents a term may appear in. $IDF(t) = \log \frac{1 + |d|}{|dt|}$ (2)

where, d is the list of document, and dt is the set of documents containing term t.

While the rare term IDF is high, whereas the IDF of a frequent term is likely to below. IDF values are calculated in order to research field and RDF graph node. The sample of IDF vector for all terms is shown in Table 1. The minimum IDF value and maximum IDF value are 0.05 and 3.09 respectively.

detection	engine	snort	Genetic	algorithm	presents	Design	principles	hybrid	intrusion
0.89	1.48	3.09	1.3	0.1	0.45	0.45	2.4	1.3	1.99

Table 1:- IDF Vector for all Terms

C. TF-IDF Weighting

The term frequency–inverse document frequency (TF-IDF) is a numerical statistic representing how important a word in a list or corpus is to a document. The value of TF-IDF increases proportionally with the number of times a term appears in the document, but is compensated by the frequency of the word in the corpus, which helps to monitor whether certain words are more common in general than others.

TF-IDF can be used effectively in the filtering of stop-words in specific subject fields like text description and classification. The weighting scheme TF-IDF assigns a weight to the term t in document d given by

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \quad (3)$$

TF-IDF weight values are calculated in according to research field and RDF graph node. The sample of TF-IDF weight matrix is shown in Table 2. The minimum TF-IDF weight value and maximum TF-IDF weight value are 0 and 3.09 respectively.

Document/Terms	detection	engine	genetic	algorithm	design	hybrid	Intrusion
D1	0.89	1.48	1.3	0.1	0.45	1.3	1.99
D2	0	1.48	1.3	0.1	0.45	1.3	0
D3	0	1.48	0	0.1	0.45	1.3	0
D4	0.89	0	1.3	0.1	0.45	1.3	0
D5	0	0	0	0.1	0.45	0	0
D6	0.89	1.48	1.3	0.1	0.45	1.3	1.99
D7	0.89	0	0	0.1	0.45	0	0
D8	0.89	0	0	0.1	0	0	0
D9	0.89	0	0	0.1	0	0	0

Table 2:- TF-IDF Weight of Terms from Abstract of RDF Graph Nodes

VI. CONSTRUCTING VIRTUAL DOCUMENTS

The RDF graph model plays an important role in the development of Semantic Web ontologies, which defines the meanings of formulations and operations for virtual document creation. A virtual document is a node of RDF graph which compromises the fields such as virdocid, authors, email, organization, paper title, research field, abstract, references with other papers, terms in the paper and associated node id of the RDF graph. The algorithm of constructing virtual documents is described in Fig 3.

Input: RDF Data Graphs
Output: Virtual Documents
Begin
1. Analyze the RDF graph data
2. Extract <rdf:Description> ….</rdf:Description> from whole document
3. Extract <paper:PAPERID>….</paper:PAPERID> assign to virdocid
4. Extract all references of virtual document
5. Extract <paper:AUTHORS>….</paper:AUTHORS>
6. Extract <paper:EMAIL>….</paper:EMAIL>
7. Extract <paper:ORG>….</paper:ORG>
8. Extract <paper:TITLE>…. </paper:TITLE>
9. Extract <paper:RESEARCH>….</paper:RESEARCH>
10. Extract<paper:ABSTRACT>…. </paper:ABSTRACT>
11. Calculate Term-Term weight matrix.
12. Find associated terms from RDF related properties.
End

Fig 3:- Algorithm of Constructing Virtual Document

ISSN No:-2456-2165

VII. SEMANTIC SEARCHING

The advanced IR methods will give a new dimension to the task of searching for huge RDF graphs. A complementary approach to creating a semantic index for a large RDF graph is proposed, based on word space model. Traditionally, a semantic index measures the similarity of words in a large set of documents based on their contextual distribution, and the similarity between documents based on the similarities of the terms contained within. By constructing a semantic index for an RDF graph, we are able to recognize contextual similarities between graph nodes (e.g., URIs and literals) and, on that basis, between arbitrary subgraphs. These similarities can be used to find a ranked list of related URIs / literals for any given input word (a literal or a URI), which can then be used to search the repository or to enrich SPARQL queries.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX j.0: <http://ucsm.edu.mm/conference/2013#>
PREFIX paper: <http://www.ucsm.edu.mm/conference/paper#>
SELECT ?PAPERID ?TITLE ?AUTHORS WHERE {?x paper:TITLE ?TITLE.
?x paper:AUTHORS ?AUTHORS.
?x paper:RESEARCH \"Artificial Intelligence\" .}
```

Fig 4:- A SPARQL Query Example

VIII. SEMANTIC SIMILARITY

Most of the international conference papers are categorized by research fields. Accessing Research paper is efficiently needed in literature survey. Papers from some of international conferences are downloaded and created RDF Graph by using Jena Framework. This system can produce all associated nodes using semantic search rather than keyword search.

Similarity measures between objects are of central importance for the various methods of data analysis. In the special case of similarity measures relating to maps, such techniques hold. Cosine similarity is one of the most common similarity measure applied to text documents, for instance in numerous requests for retrieval of information. We calculate the degree of similarity of two documents with documents presented as vectors as the correlation between their respective vectors, which can be further quantified as the angle cosine between the two vectors.

The similarity measures map the distance or similarity between two objects' symbolic descriptions into a single numerical value, which depends on two factors- the properties of two objects and the measure itself. Two documents $\vec{t}_a$ and $\vec{t}_b$, their similarity is

$$SIM(\vec{t}a, \vec{t}b) = \frac{\vec{t}a.\vec{t}b}{|\vec{t}a| \times |\vec{t}b|} \quad (3)$$

where $\vec{t}_a$ and $\vec{t}_b$ are m-dimensional vectors over the term set T = {t1, . . . , tm}.

PaperId	r215	r216	r151	r152	r425	r148	r147
r215	1	0.16	0.27	0.02	0.17	0.03	0.11
r216	0.16	1	0.11	0.02	0.20	0.017	0.053
r151	0.27	0.11	1	0.06	0.09	0.031	0.024
r152	0.02	0.02	0.06	1	0.007	0.008	0.040
r425	0.17	0.20	0.09	0.007	1	0.102	0.052
r148	0.03	0.01	0.03	0.008	0.102	1	0.163
r147	0.11	0.05	0.02	0.04	0.052	0.163	1

Table 3:- Similarities of web engineering Resources and other resources

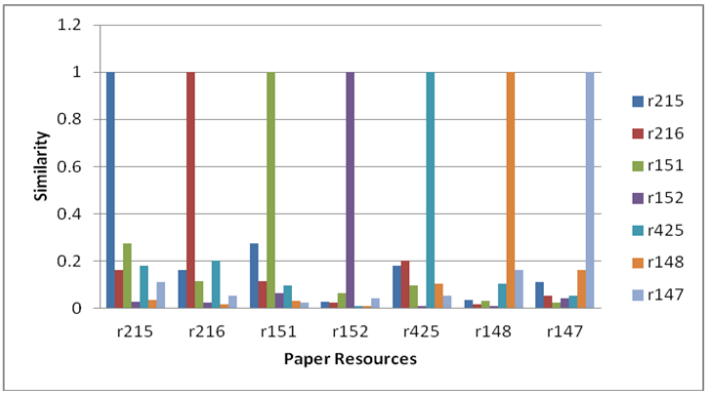

Fig 5:- Similarities of Web Engineering Resource and Other Resources

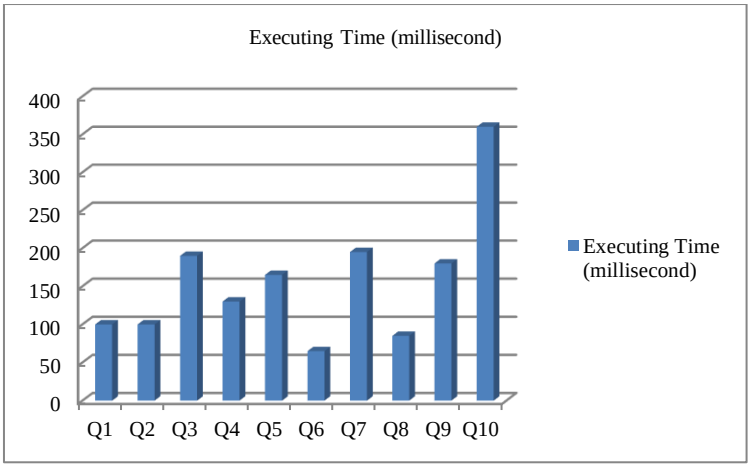

Fig 6:- Executing Time for Sample 10 Queries

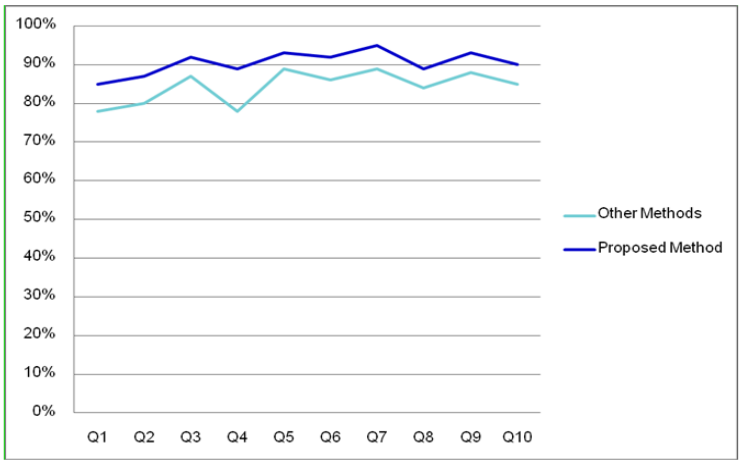

Fig 7:- Comparison of User Satisfactory Proposed Method and Other Methods

The Fig 7 shows the comparison of user satisfactory rate of semantic searching between proposed method and other methods. It explains that our proposed method provide high user satisfactory rate than other methods. In order to achieve the semantic searching, the proposed method can produce the relevant answers. By checking above the queries, we find that the proposed approach retrieves the better applicable results than the current methods.

IX. CONCLUSTION

The main intention of this system is to exploit the best features for semantic search from both Information Retrieval and Semantic Web. With the ultimate goal of improving efficiency of conventional keyword-based search, this paper proposes creating a semantically enhanced model of search over RDF Graph. It incorporates and utilizes highly formalized semantic information in the form of RDF and Knowledge Base (KB) within conventional IR ranking models.

Starting from this position and trying to bridge the gap between these two groups, this paper examines the idea of an IR model based on RDF, aimed at exploiting the KB domain. This paper combines the benefits of both keyword search and semantic search. The semantic search's primary objective is to improve conventional IR systems that are based on word computation. We are committed to researching the use of the Semantic Web technologies to the mission of knowledge recovery.

REFERENCES

[1]. C. Bizer, T. Heath, T. Berners-Lee: "Linked Data - The Story So Far", To appear in: *International Journal on Semantic Web and Information Systems (IJSWIS), 2010.*
[2]. D. Brickley, and R.V. Guha. "2014. RDF Schema 1.1. W3C Recommendation 25 February 2014".
[3]. E. Marx, et al. "Exploring Term Networks for Semantic Search over Large RDF Knowledge Graphs", *Semantic Web 1 (2017) 1-5*.
[4]. E. Shady, R. Maya, S. Ralf, W. Gerhard. "Searching RDF Graphs with SPARQL and Keywords", *Bulletin of IEEE Computer Society Technical Committee on Data Engineering*, 2010.
[5]. E. Prud'hommeaux, and A. Seaborne, "SPARQL Query Language for RDF, W3C 2008".
[6]. F. Suchanek, G. Kasneci, G. Weikum: "YAGO: a Core of Semantic Knowledge". WWW 2007

[7]. G. Cheng,W. Ge, and Y. Qu. Falcons: "Searching and browsing entities on the Semantic Web", *In Proc. WWW-2008, pp. 1101– 1102, ACM Press, 2008*.
[8]. G. Kasneci, F. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. "NAGA: Searching and ranking knowledge". *In Proc. ICDE-2008, pp. 953–962, IEEE Computer Society, 2008*.
[9]. G. Salton and M. McGill, "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.
[10]. G. Tummarello, R. Delbru, E. Oren: "Sindice.com: Weaving the open linked data". *In: Proceedings of the 6th International Semantic Web Conference*. Busan, Korea (2007)
[11]. J. Breslin, A. Passant, S. Decker: "The Social Semantic Web", Springer, 2009
[12]. K. Anyanwu , A. Sheth (2003) "ρ -Queries: Enabling Querying for Semantic Associations on the Semantic web", *Proc of the 12th International World Wide Web Conference. ACM Press* pp 690–699
[13]. M. Aydar, S. Ayvaz, and A. C. Melton. "Automatic weight generation and class predicate stability in rdf summary graphs", *In Workshop on Intelligent Exploration of Semantic Data (IESD2015),* co-located with ISWC2015, 2015.
[14]. P. Castells, M. Fernández, and D. Vallet. "An adaptation of the vector-space model for ontology-based information retrieval", *IEEE Trans, Knowledge Data Engneering*, 19(2):261–272, 2007.
[15]. S. Paliwal et al. **"**Constructing Virtual Documents for Keyword Based Concept Search in Web Ontology", *International Journal of Engineering and Technology (IJET)*, pp. 1347–1354, ISSN: 0975-4024, Vol 5 No 2, Apr-May 2013.
[16]. Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu. "Spark: Adapting keyword query to semantic search", *In Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007),* November 2007. Springer Verlag.
[17]. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. G. Ives: "DBpedia: A Nucleus for a Web of Open Data", ISWC/ASWC 2007
[18]. T. W. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. "Swoogle: Searching for knowledge on the Semantic Web", *In Proc. AAAI-2005*, pp. 1682–1683, AAAI Press / MIT Press, 2005.
[19]. Y. Qu, W. Hu, G. Cheng.: "Constructing virtual documents for ontology matching". *In: Proceedings of WWW2006*. pp. 23{31 (2006)