VIRTUALIZING DOCUMENT ALGORITHMS USING PREDICTIVE SEMANTIC DATA

Dr. MUHAMMAD USMAN TARIQ
Assistant Professor, Abu Dhabi School of Management, Abu Dhabi, UAE

ABSTRACT

Semantic web technologies play a vital role in enhancing real-world applications. With the advent of time, information is readily available on the internet in various formats, including files, metadata documents (Microformats, RDF, RDFa), and documents. Often traditional search methods do not offer the adequate and required level of matching users’ information with the available online documents, which act as a barrier for efficient usage and reproduction of adapting keywords. This research focuses on an approach that automatically translates user-provided queries into the required formal structured queries. Users can use the approach to perform the translation efficiently. Moreover, the research focuses on the construction of a virtual document and queries for the semantic web data. Other than this, a more advanced search interface with a keyword-based approach is introduced for searching and retrieving most relevant objects.

KEYWORDS: SPARQL; RDF graph; Virtual documents; Semantic search & Semantic Data

Received: May 05, 2020; Accepted: May 25, 2020; Published: Jun 18, 2020; Paper Id.: IJMPERDJUN2020202

1. INTRODUCTION

Data containing the various form of information over the internet is increasing the rapid demand for methods, tools, and techniques to handle a large amount of semantic web data [1]. Current approaches mostly focus only on the keyword-based search for finding relevant data based on random word generation. Such models do not focus on analyzing the actual metadata information and rely on the hit and trial method for matching keywords that do not relate with the actual semantic information of the textual data [9]. The challenge of mismatching keywords can be minimized by using ontologies and resource description framework (RDF) for visualizing the required knowledge that can act as a backbone of the web semantic applications. Such approaches provide minimum set semantics to the plain text, which can be beneficial for searching, extraction, and matching process based on metadata. Several query languages are designed to focus on semantic queries; out of all, SPARQL provides advanced query language for searching the semantic web. Based on the RDF and ontologies knowledge, the next step is to work on semantic knowledge.

Data available on the internet consists of different types of data having structured and semi-structured data that is readily published by a large number of users, which are both individuals and organizations. Various semantic annotations for the extensive and mixed resources are available in RDF format, which is being searched using the RDF language called Simple Protocol and SPARQL Query Language (SPARQL). SPARQL is a query language that focuses on information retrieval based on new and old data that are available in RDF format. The representation in RDF format increases the feasibility of creating mapping among RDF data that originates from primary traditional sources and creates big RDF repositories. The increasing daily amount of large amount of structured data is also available in RDF triple format. The semantic RDF has gained popularity for scientific data [13], for example, Web 2.0 applications [9], DBpedia [13], Yago [4], and Wed of Data [1]. RDF models the databases as graphs by creating
entities in the knowledge base with guided paths consisting of linked entities. The RDF data collection can contain triple property objects and SPOs. SPO triple is defined as a pair of entities that can convert the other triples and can also be viewed as a tree of different written edges and nodes that can have billions of triples. The node consists of edges and entities of relationships [2].

Often the RDF graphs face inconsistencies, incompleteness, and redundancies due to different types of methodologies for the generation of RDF graphs. These graphs are not fully covered to ensure the quality of data on a high level and searching through the dense RDF graphs, which require in-depth knowledge of structure and knowledge of correctly used elements. A large amount of work has been done for exploration, and retrieval of RDF triples and existing approaches support users in the search process. However, it lacks full support for the required level of semantic search. Additionally, the information retrieval method can provide new perspectives for information retrieval on the large RDF graphs based on improving the semantic search abilities.

2. RELATED WORK

Keyword-based ontology search engines are based on exploring ontology concepts, a combination of terms, and popularity scores. One of the keyword-based ontology search engines is known as the falcon search [5]. Falcon search indexes all the keywords from the virtual document to classes, properties, and uses an inverted index from terms to concepts. It selects the top-ranking classes and dynamically groups them by their names for recommending to users. The term-based system provides ranking scores of virtual documents using keyword queries. Another system is named Swoogle [14] that is used as a crawler system for indexing, querying, exploring, and searching RDF documents. It mainly focuses on providing semantic web searches and documents. Swoogle uses indexing based on extracted keywords from the uniform resource identifiers (URIs) against the RDF files. It allows users to provide queries that link to the metadata of the documents which have RDF extension. It further allows users for the semantic search of web documents having RDF/XML language as syntax. Sindice [8] provides search centralized around entities and query service for already indexed web data ranking according to the popularity of the associated keywords of the data set. Sindice functionality is based on collecting RDF documents and work on indexing the keywords and URIs as resources. Further, it allows users to explore the resources associated with different URIs and identify similar documents using the inverted index scheme. Another popular search engine is a NAGA semantic search engine [6]. It is based on querying language based on graphs language for searching the underlying knowledge of files where the knowledge base is represented as a graph. It enhances the knowledge base by extraction of web sources, where nodes and edges are as represented as graph The NAGA search engine extends SPARQL, which allows us to search complex graphs using regular expressions rather than edge relationships. Another retrieval system that is based on SPARQL is SPARK [12] that can query semantic data based on the given keywords. SPARQL is based on three steps that include constructing graphs, ranking queries, and mapping keywords. Ontological resources are matched with the query keywords. The construction of the graph focuses on analyzing user given queries for the development of missing relationships to obtain SPARQL queries. The last step focuses on the queries relevant to the model ranking based on the highest rate of SPARQL queries.

3. BACKGROUND
3.1 RDF Data Model and SPARQL

The RDF model and SPARQL model can be explained through representation of pairwise disjoint infinite set or internationalized resource identifiers (IRIS), literals, variables, and blank node. For this purpose, we have taken four nodes named as \( A_1, B_1, C_1, \) and \( D_1 \) as pairwise. \( A_1B_1, A_1C_1, A_1B_1C_1, \) and \( A_1D_1C_1 \) represent \( A_1 \cup B_1, A_1 \cup C_1, A_1 \cup D_1, A_1 \cup B_1 \cup C_1, \) and \( A_1 \cup D_1 \cup C_1 \) respectively. RDF terms can be defined as \( A_1B_1C_1 \). The tuple for the RDF triple is defined as “\( T^{i} \)” \( (x, y, z) \in (A_1B_1) \times A_1 \times (A_1B_1C_1). \) The subject is defined by \( x \), predicate is defined by \( y \), and object is defined by \( z \). The RDF graph \( G \) can contain set of RDF triples. An example of RDF graph is shown is figure 1, that consists of set of triples, labeled graph, edges showing subject to object, predicates, circles shows IRIs, and rectangles showing literals. The triple is defined as \( T^{i}P^{i} \) having \( (xy, yy, zy) \in (A_1C_1D_1) \times (A_1D_1) \times (A_1D_1C_1). \) The subject pattern is defined as \( xy \), predicate pattern as \( yy \), and object pattern as \( zy \).

![Figure 1: Sample RDF Graph.](image)

3.2 Semantic Associations

The semantic associate shows relationships among complex resource-based entities. The effective association will be showing fewer complex relationships of medium level entities, which enable users to establish links easily between different points, events, people or places [10]. Different association definitions are used for semantic connectivity and similarity. If there is a sequence between RDF graph and entities, they are semantically connected whereas if there is path between two entities; they are semantically similar.

4. 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 the 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. List of tuples are matched through the SPARQL query against the knowledge base (KB). There are three main components in the semantic search: semantic analysis, semantic search, and model creation. Our new approach provides an extended key-based retrieval system with semantic knowledge base. Figure 2 shows design architecture of searching semantic web data through RDF graph.

[www.tjprc.org]  [SCOPUS Indexed Journal]  [editor@tjprc.org]
5. SEMANTIC ANALYSIS

The semantic analysis is based collecting the words which are gathered through though the textual content analysis of the RDF document. The collection of words is given a weight and named and assigned a new dimension of the vectors for prioritizing the document. The words are normalized during the preprocessing step for elimination of stop words and stemming. The virtual document contains the weighted rational numbers. A non-zero weight is assigned to the value in the vector if a document includes the required term. Another common method of inverse document frequency is often used to compute the weights. The weights for each word are increased according to the relevant number of term appearances in the documents by scaling down the frequency of terms 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. While the standard vector-based search is based only on the presence and frequency of terms in documents and does not consider any additional information about terms themselves. Terms from the title and abstracts that extract RDF Data Graph are inverse document frequency, calculated weighting, maximum term frequency, nominal term frequency, and normalization.

5.1 Nominal Term Frequency
The nominal term frequency has collection of documents \( \{F_1, \ldots, F_n\} \) and \( \{N_1, \ldots, N_m\} \) as collection of different terms in a document with a m-dimensional vector. The frequency of the term is denoted by \( R \in T \) of a document \( W \in A \). The vectors in the document are represented as

\[
TR(R,W) = 1 + \log(1 + \log(\text{freq}(R,W)))
\]

Frequency of each term is counted in the document and allocated a weight according to the number of frequencies of each term. Score between the term \( R \) and document \( W \) is calculated based on the weights of \( R \) and \( W \). Another simple method is to allocate the weights equal to the frequency of term in a document; knows as \( TR(R,W) \) with the subsequent terms in the same order. The numerical digest of the document is based on the weights of \( TR \). The data of the document is stored according to the frequency of each term. Commonly the terms are just words. First, the stop words are removed, which are mostly non-descriptive words. Table 5.1 provides specific research fields and their term frequency. The term frequency of each RDF graph node is calculated from our RDF graph data model. The minimum term frequency and maximum term frequency for each RDF graph node are 24 and 85, respectively.

### 5.2 Inverse-Document Frequency (IAF)

Inverse document frequency (IAF) is based on calculating the value of a term in the text documents. The number of documents stored in the collection containing the term \( t \) is measured as total document frequency (aft). The measured value on document and term level is differentiated by using document level statistics. The differentiation is necessary as document frequency (df) is behaves differently than the term frequency (cf). Particularly, cf values for the try and insurance are usually equal but there is huge difference in the df values. Insurance is purposely used to get a higher boost for query insurance to get higher hits for query on insurance rather than just having large set of documents containing try from query of try. The collection of documents is denoted by “m” and IAF by “d” as:

\[
\text{IAF}(d) = \log \left( \frac{1 + |m|}{|md|} \right)
\]

IAF of frequent terms is usually low whereas highly rare terms is high. IAF values are calculated in order to research field and RDF graph node. The sample of IAF vector for all terms is shown in table 1. The minimum IAF value and maximum IAF value are 0.05 and 3.09, respectively.

### 5.3 TF-IAF Weighting

The combination of the term frequency and IAF is sued to calculate the composite weight of each term in every document know as term frequency document frequency (TF-IAF). TF-IAF helps to understand and reflect the importance of a word to a document in corpus or collection. It is further used for text mining, information retrieval, increasing the value proportional to the frequency of words in a document. Additionally, the frequency of the words in the corpus or document in a collection is managed with offset frequency to control the frequent words usage. Search engines use the common variations of TF-IAF as a tool for central ranking and scoring the documents closest to the user query. TF-IAF is also used for measuring and filtering stop words in various areas including classification and summarization of text. The weights in the TF-IAF for a term “P” in a document “D” is as follows:
\[ \text{TF-IAF}(P,D) = \text{TF}(P,D) \times \text{IAF}(P) \]

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

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

| Document/Terms | Detection | Engine | Genetic | Algorithm | Design | Hybrid | Intrusion |
|----------------|-----------|--------|---------|-----------|--------|--------|-----------|
| A1             | 0.89      | 1.48   | 1.3     | 0.1       | 0.45   | 1.3    | 1.99      |
| A2             | 0         | 1.48   | 1.3     | 0.1       | 0.45   | 1.3    | 0         |
| A3             | 0         | 1.48   | 0       | 0.1       | 0.45   | 1.3    | 0         |
| A4             | 0.89      | 0      | 1.3     | 0.1       | 0.45   | 1.3    | 0         |
| A5             | 0         | 0      | 0       | 0.1       | 0.45   | 0      | 0         |
| A6             | 0.89      | 1.48   | 1.3     | 0.1       | 0.45   | 1.3    | 1.99      |

6. VIRTUAL DOCUMENTS CONSTRUCTION

Semantic web ontologies foundation has an RDF graph model as an important part of them. It contains the definitions of the operations, formulations, and virtual documents construction. A virtual documents consists of the RDF graphs and RDF triples which are in form subject, predicate, and object. The literals in the documents are made up of identification forms without differentiation properties and with a node in the 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 for constructing virtual documents is described in figure 3.

Input: RDF Data Graphs
Output: Virtual Documents

Begin
1. Analyze the RDF graph data
2. Extract \texttt{<rdf:Description> \ldots</rdf:Description>} from whole document
3. Extract \texttt{<paper:PAPERID> \ldots</paper:PAPERID>} assign to virdocid
4. Extract all references of virtual document
5. Extract \texttt{<paper:AUTHORS> \ldots</paper:AUTHORS>}
6. Extract \texttt{<paper:EMAIL> \ldots</paper:EMAIL>}
7. Extract \texttt{<paper:ORG> \ldots</paper:ORG>}
8. Extract \texttt{<paper:TITLE> \ldots</paper:TITLE>}
9. Extract \texttt{<paper:RESEARCH> \ldots</paper:RESEARCH>}
10. Extract \texttt{<paper:ABSTRACT> \ldots</paper:ABSTRACT>}
11. Calculate Term: Term weight matrix.
12. Find associated terms from RDF related properties.

End

Figure 3: Algorithm of Constructing Virtual Document

7. SEMANTIC SEARCHING
The similarity of the terms is based on the semantic index and contextual distribution in a vast set of documents collection. The similarities can be measured for the inside and outside terms in a document. Contextual similarities among the graph nodes can be analyzed through creation of semantic index that can help to find a rated list of literals for the user given inputs terms using SPARQL queries. A SPARQL query example is shown in figure 4.

![Figure 4: A SPARQL Query Example](image)

### 8. SEMANTIC SIMILARITY

Most of the international conference papers are categorized by research fields. Accessing research paper is efficiently needed in the literature survey. Papers from some of the international conferences are downloaded and created an RDF Graph by using the Jena Framework. This system can produce all associated nodes using semantic search rather than a keyword search. The measure of proximity between two natural language terms is calculated through the semantic proximity. The semantic similarity is matched with the available dataset of terms by using the level of semantic relatedness. Further, similarity measure is used to detect the reduce class of semantic relations by calculating the similarity between different terms. Similarity measures are important for data analysis of the different objects and the graphs. The documents are represented as term vectors for measuring the similarity of two documents based on correlation between two vectors. It is measured through cosine similarity and angle between vectors that is also applied to the textual documents that further reflects the degree of relatedness or degree of differentiation. The similarity of the two different objects is quantified into a single value depending on the properties of objects and the measure itself. A similarity between two documents $d_S$ and $d_Z$ is shown as:

$$SIM(d_S,d_Z) = \frac{d_S \cdot d_Z}{|d_S| \times |d_Z|}$$

where $d_S$ and $d_Z$ are vectors of m-dimension over the term set $D = \{d_1, \ldots, d_m\}$. The similarity of document resources and dimensions of the vectors for current research documents is between [0,1] and non-negative as shown is table 3.

| Paper Id | r215  | r216  | r151  | r152  | r425  | r148  | r147  |
|----------|-------|-------|-------|-------|-------|-------|-------|
| r215     | 1     | 0.160607 | 0.273825 | 0.026215 | 0.179329 | 0.033607 | 0.110203 |
| r216     | 0.160607 | 1     | 0.114611 | 0.022453 | 0.202647 | 0.017653 | 0.053623 |
| r151     | 0.273825 | 0.114611 | 1     | 0.062597 | 0.095166 | 0.03186 | 0.02411 |
| r152     | 0.026215 | 0.022453 | 0.062597 | 1     | 0.007869 | 0.008346 | 0.040533 |
| r425     | 0.179329 | 0.202647 | 0.095166 | 0.007869 | 1     | 0.102655 | 0.052929 |
| r148     | 0.033607 | 0.017653 | 0.03186 | 0.008346 | 0.102655 | 1     | 0.163091 |
| r147     | 0.110203 | 0.053623 | 0.02411 | 0.040533 | 0.052929 | 0.163091 | 1     |
Figure 4: Similarities of Web Engineering Resource and Other Resources

Figure 4 presents the similarities of the web resources into a graphical format which shows the results for comparison in more efficient manner and clarity of the range which is between [0,1] and with other resources which are available on the web.

Figure 5: Executing Time for Sample 10 Queries

Figure 5 shows the executing time for 10 sample queries using the SPARQL query format which is in range of [0, 350ms] which shows a fast processing using the current methodological approach.
Figure 6: Comparison of User Satisfactory Proposed Method and Other Methods

A comparison of a satisfactory user rate of semantic searching between other methods and the proposed method is shown in figure 6. The results show that our proposed method has high user satisfaction than other models. The proposed method can provide relevant solutions for semantic searching. The results show that the proposed method has fast research results and query processing than the other methods. Overall, the approach is more efficient for semantic web searching and analyzing the queries.

9. CONCLUSIONS

The current research focuses on analyzing the gaps between the IR and software development communities in realizing the importance of semantic search for better understanding of virtual document queries. The current research has proposed the development of an advanced semantic search model that exploits and combines high formalized semantic knowledge in RDF graph and knowledge base. The research has proposed the model within the traditional models of semantic web search. The purpose is to improve the conventional IR approach based. The results of the research show that the proposed model has higher query processing rates as well as satisfaction rate over the other proposed models of semantic web search. The further research can be based on using the current research model for a range of keywords within one document to better understand the retrieval system.

Conflict of Interest: We have no conflict of interest with anyone or any party for this paper.

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. 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.

3. E. Prud’hommeaux, and A. Seaborne, SPARQL Query Language for RDF, W3C 2008. http://www.w3.org/TR/rdf-sparql-query/

4. F. Suchanek, G. Kasneci, G. Weikum: YAGO: a Core of Semantic Knowledge. WWW 2007

5. 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.
6. 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.
7. G. Salton and M. McGill, Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
8. 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)
9. J. Breslin, A. Passant, S. Decker: The Social Semantic Web, Springer, 2009
10. K. Anyanwu, A. Sheth (2003) p-Queries: Enabling Querying for Semantic Associations on the Semantic web, Proc of the 12th International World Wide Web Conference. ACM Press pp 690–699
11. P. Castells, M. Fernández, and D. V allet. An adaptation of the vector-space model for ontology-based information retrieval, IEEE Trans, Knowledge Data Engineering, 19(2):261–272, 2007.
12. 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), Busan, South Korea, volume 4825 of LNCS, pages 687–700, Berlin, Heidelberg, November 2007. Springer Verlag.
13. 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
14. 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.
15. Y. Qu, W. Hu, G. Cheng.: Constructing virtual documents for ontology matching. In: Proceedings of WWW2006. pp. 23(31 (2006)
16. 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.
17. E. Marx , et al. “Exploring Term Networks for Semantic Search over Large RDF Knowledge Graphs”, Semantic Web 1 (2017) 1–5
18. Brickley, D. and Guha, R.V. (eds). 2014. RDF Schema 1.1.W3C Recommendation 25 February 2014. https://www.w3.org/TR/rdf-schema/
19. Danthala, S. W. E. T. H. A., Rao, S. E. E. R. A. M. S. R. I. N. I. V. A. S. A., Manepalli, K. A. S. I. P. R. A. S. A. D., & Shilpa, D. (2018). Robotic Manipulator Control by using Machine Learning Algorithms: A Review. International Journal of Mechanical and Production Engineering Research and Development, 8(5), 305-310.
20. Kannemadugu, R., & Devi, A. L. Performance Improvement Of Transmission System Using Upfc By Ga And Pso Algorithms.
21. Madhuri, D. M. S., Annapurna, G., Venkataramana, C., & Swetha, G. Text Hiding Using Rsa And Blowfish Algorithms With Hash-Based Lsb Technique.
22. Jaladi, S., & Rao, T. An Inverse Kinematics Analysis Of Space Station Remote Manipulator System (Ssrrms) Using Genetic Algorithms.