Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 490 – 495

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Keyword Prediction with ARM on Bibliographic RDF Data

Nidhi Kushwaha, Bharat Singh, Rajesh Mahule, O P Vyas

Indian Institute of Information Technology, Allhabad, India

Abstract

Web-3.0 provides an easy way to utilize the in-depth knowledge of the huge data that grows day-by-day in the internet Our aim with this paper is to work with the Linked Open Data Cloud data, where the main problem with the dataset is inconsistencies, bulkiness. We are exploring bibliographic data which is one of the cloud data. The authors found some useful information in the dataset that should be explored for judging the improvement of the search query's result. After analysis we came to know that many of the papers residing in RKBExplorer did not have keyword information. Because of that the search engine based on the RKBExplorer only able to use the information in this database going to retrieve the papers, authors of that paper and their related cited papers with given paper author or title. But assume the situation where the user wants to enter the search string, then what would be the result? Would it retrieve all the related paper even if their keywords are not assigned? In this paper we are trying to answer this question, with the help of data mining algorithm ARM on the features retrieved from the RDF data. We have developed a novel approach through which we can answer the user's query which is mixture of important the strings, we called them tags of the papers

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Keywords: Linked Open Data Cloud; Data Mining; Query-Answering System; RDF; Assoication Rule Mining.

1. Introduction

Linked Open Data (LOD) Cloud Projects [1] had started in the year 2008 by Tim Berners-Lee. It comes with the idea of open source sharing of information on the web, which globally connects the data using unique URIs. Publishing the data gives an opportunity to the universities and researchers to load the data for the internet users in various application domains. The Google Rich snippet and Yahoo Search monkey show the good example of embedding information of RDF data into the less informative XML documents. Today's various applications and browsers support RDF data. This shows its continuous growth and usefulness with the current working environment. Traditional web consists millions of pages connected with each other. But the logic behind the connectivity was missing. This causes the problem to connect the future relatedness of the documents. Ontology development needs a specialized person who has good knowledge of that field. Because of domain

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.019

dependent these developed Ontologies had different connections and concept names [10]. Now the problem was how to combine them, it has been said that use the well known predicates for newly developed Ontology can reduce this problem. One of the way found by Ontology engineers was the use of ontology development tools and the important predicate link known as "owl:sameAS". Linked Open Data Cloud, a continuously growing cloud done this work under some norms defined in [1][10]. The Cloud has various domain information including a cross domain giant "DBpedia" [1]. Many universities and organizations come forward to success the dream of Sir Tim Berners-Lee (in 2006). The cloud itself presents an example of the diverse information sources. Various works have been going on to connect this diverse information. Consider an example, the person is related to FOAF ontology may also connect with the DBLP paper (DBLP Ontology) with the has-author relationship. The co-author of the paper might be taught in the same university (University Ontology) who's one of the student presents this paper in the conference (Event Ontology) held in Japan (Geoname Ontology). Former example explained the connectivity of a person to another, to a paper, to an organization, and to the country itself. Accessing all this information automatically without moving through hyperlinks of the pages was the initial idea of the LOD Cloud. Linked Open Data Cloud, a phenomena of Tim Berners Lee already taken a well established space in the current applications [1][2][3]. Many organizations have come forward in the last decades to provide successful open source Knowledge Base [1]. This knowledge has been utilized in many applications [4] to give the meaningful results by combining different data sources. This was only possible because of their same structured format. The triplet of RDF contains information about the concepts in the form of their relationship among all other but related concepts. These links also have some special characteristics. Different links can attach with different objects or value. Like an actor of the movie can be a director also. But this is not the case when we are taking about bibliographic database. RKBExplorer [5] contain information about bibliographic data, we explore the data for generating tags from it. RKBExplorer provides the unified view of the heterogeneous data sources. ReSIST project [5] proposed a semantically enabled knowledge structure. The aim of the project was to provide services from different but related data sources. In the next sections we describe about methodology, implementation and the results of the framework, finally concludes with future work in the last section.

2. Triplet Extraction through multiple RDF datasets

We have studied 3-Bibliographic datasets named as: DBLP, IEEE and ACM. These are the very basic and most utilizes datasets in the bibliographic searching. Users in this search are not ordinary users they are trained enough to utilize the result of the searching queries [9]. But the time consumption for utilization of these searches is the main problem in this. Linked Open Data Cloud is a good example of semantic connectivity among the huge knowledge source. The information provided by the datasets in this cloud is semantically linked with each other. We can consider the datasets as a huge graph in which the vertices are the subjects and objects.

Linkage information is consumed as a predicate unique. between the subject and object. So in-conjunction, the information is called as a triplet. The linkage information gives us an opportunity to specifically utilize the objects or value. Here the authors have listed some interested information about the three RDF datasets. Thanks to the bibliographic RDF converter organization that provide a common ontology for all the three datasets. Observation told that some common information like "sub-area-of", "has-author", "fullname", "has-title", "has-date", "year-of" presents in the three data sets(IEEE,ACM,DBLP).

Some information has similar meaning but different predicates are used to them like "cites-publication-reference" in DBLP, ACM and "is-very-strongly-related-to", "is-strongly-related-to", "is-related-to" information in IEEE. Another example of this is "has-ieee-keyword" of IEEE and "address-generic-area-of-interest" in ACM dataset.

In our discussion we called these similar terms as the complementary terms (purple color). Our aim to utilize the complimentary terms as well as direct information of the 3 datasets with the least preprocessing steps for the ease of the use it.

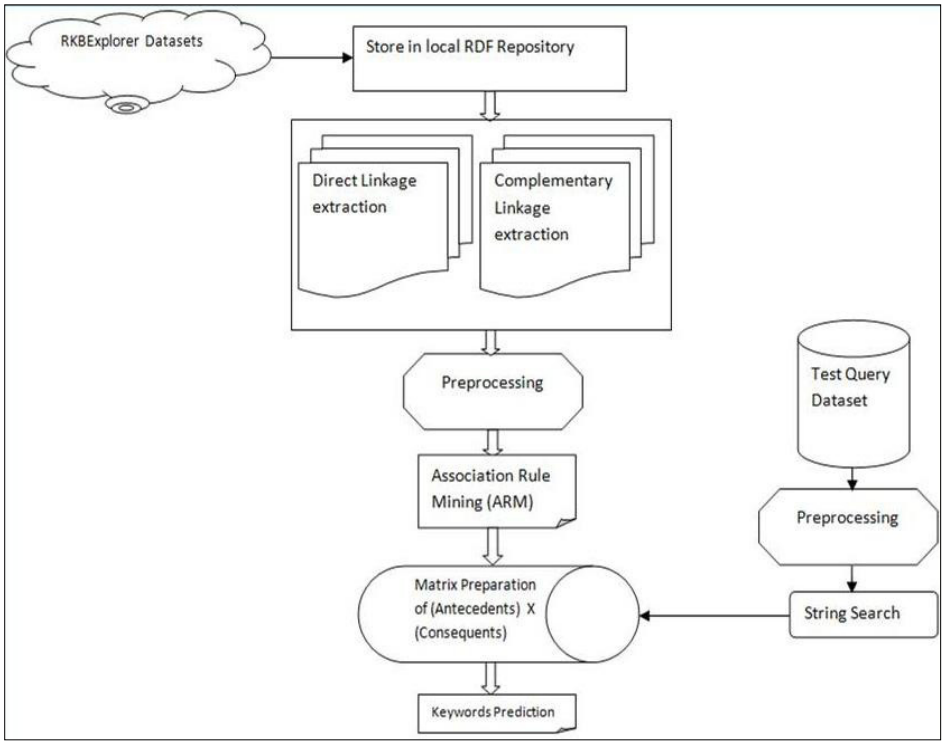

Fig. 1. Framework of Keyword extraction from Bibliographic RDF data.

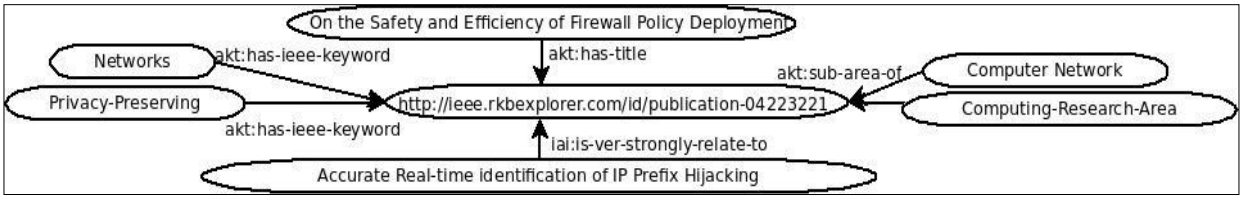

Fig. 2. IEEE-RDF graph excerpt

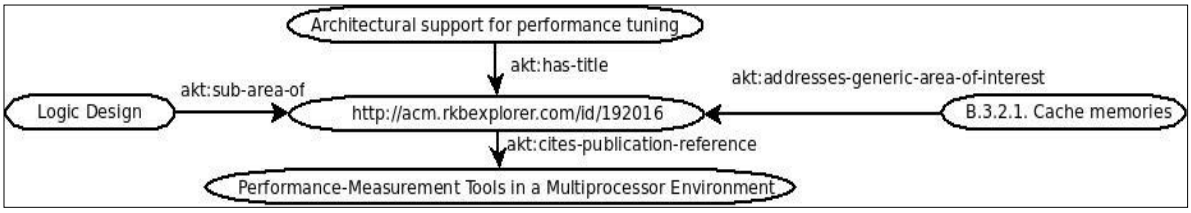

Fig. 3. ACM-RDF graph excerpt

Table.1. Predicates of the DBLP/IEEE/ACM Datasets

DBLP	IEEE	ACM
akt:sub-area-of	akt:sub-area-of	akt:sub-area-of
akt:article-of-journal	iai:is-strongly-related-to	--
akt:has-author	akt:has-author	akt:has-author
akt:full-name	akt:full-name	akt:full-name
akt:has-title	akt:has-title	akt:has-title
owl:sameAs	akt:paper-in-proceedings	--
akt:has-date	akt:has-date	akt:has-date
akt:has-web-address	akt:has-web-address	--
akt:has-volume	extn:has-abstract	--
akts:year-of	akts:year-of	akts:year-of
akts:month-of	--	akts:month-of
---	iai:has-ieee-keyword akt:has-ieee-keyword	akt:addresses-generic-area-of-interest
akt:has-affiliation	--	--
akt:cites-publication-reference	iai:is-very-strongly-related-to	akt:cites-publication-reference
akt:edited-by	iai:is-related-to	akt:has-publication-reference

3. Proposed Algorithm

For Training:

1. Select bibliographic datasets from RKBExplorer ('D_1', 'D_2', 'D_3', ...'D_n') and store in local repository 'R'.

2. Direct & Indirect properties extraction among datasets from local server Sesame[11]:

a) Fetch the data by SPARQL1.1 querying to extract all the predicates from Di ϵ ('D_1', 'D_2', 'D_3', ...'D_n') and write the result in corresponding text, Wi .

b) In D_i<S, P, O>, where $s_i \subseteq S$, $p_i \subseteq P$, $o_i \subseteq O$ and $D_i \epsilon$ 1, 2 , 3, 4....n.

c) If pi=pj=pk then properties are said to be direct links, 'p_{dir}'.

Otherwise, If $p_i \neq p_j = p_k$ or $p_i = p_j \neq p_k$ or $p_i \neq p_j \neq p_k$ then it called as indirect links, 'P_{indir}'. Where, $p_i \epsilon D_i$, $p_j \epsilon D_j$, $p_k \epsilon D_k$.. For given table.1 D_{dblp} , D_{ieee} , D_{acm}.

3. Normalization of extracted 'Oi' related to Direct & Indirect properties, steps are following:

If (Oi ϵ Str) then,

{ If (Oi==l_i), remove the term
else If ($O_i \epsilon$ kywd), replace the space with '_' (underscore) sign
}

Here, Str denotes String values. 'l' represents stop words list, 'kywd' denotes keywords that belongs to the particular paper ID.

4. Concatenate normalized 'norOi' related to Direct & Indirect, by below steps, for a particular subject Si,

If (norOi ϵ Si) then, append Oii, Oij, Oik in one row that belongs to one instance in dataset. Similarly prepare list for all the S ϵ D_i , D_j , D_k & called it 'Ð'.

5. Apriori (Ð, Supp, Conf)

```
 For p=1 to h
     For q=1 to m
{
   Mpq =Conf(p,q)
}
Return M
```

For Testing:

Steps for obtaining test result:

$T=\{t_1,t_2,t_3....t_n\}$ a set of test queries, where ti(w1,w2,w3,...wn) and ti $\neq \emptyset$ after normalization.

If $(w_1,w_2,w_3,...w_n)$ match with Mp , where $W_i \epsilon\ t_i$ then, choose M_q, where Max(Conf($w_1,w_2,w_3,w_4,...w_n$)) and store it after concatenation in 'Reslt' list, after that deduct this 'M_q' every time from 't_i'. Until the Max(Conf($w_1,w_2,w_3,w_4,...w_n$)) <=Conf.

Display 'Reslt' list as a result set for test query 't_i'.

4. Implementation Detail of RDF storage, Extraction and Mining

For We have taken approx. 300 data from IEEE and 500 data from ACM. Because of adequate knowledge of keyword we cannot consider DBLP data. After selecting the direct and complementary features, preprocessing step will remove all the stop words and format the keyword as disused above. Now our dataset is ready for the ARM generation [6] [7]. Next step is to convert the result into the matrix of N*M, where 'N' is the Antecedent and 'M' is the consequent in the generated rule. Now we are ready to test our rules, this will be done with the preprocessing step, where we remove all the stop word presents in the datasets and then use these tokens as string search. The strings are then compared to the antecedents of the matrix and the corresponding consequents are the results. These results are selected priority wise the first result would be that whose confidence is greater than all others in the corresponding vector of matched string. For example: In table 2. If the search string is matched with the A1 then order of recommendation would be C4, C1, C3, and C2. In Fig.4, the test Query is the title of the paper whose keyword has not assigned previously.

4. Evaluation

To evaluate the proposed model we have divide the whole dataset into 75% and 25%. This dataset have RKBExplorer information of different publishing group like ACM, IEEE. Training data have 75% of whole dataset which will be used for training the model while test data have only 25% information. For our purpose we have taken total 1200 papers information for model training. To make the result more effective we first consider to prepare dataset dedicated to specific domain, in our case it is networking and information retrieval. Next, preparation of matrix M_{pq}, from which we fetch the results after preprocessing of the test query. After obtaining the result we match it with the known keywords that are already associated with the particular papers. Calculation of Precision will be done by matching the known and predicted values. Fig. shows that result of it. For evaluation we also introduce one more feature that will be judge the effectiveness of output by manual expertise and named as 'Novelty'. It shows the new keyword that was not previously attached with the specific paper but determined by our model. These novel keywords will be generated by considering both the papers that occur into the datasets it may be not present in one publishing group but the model is able to obtaining the result from others. The generated keywords that comes into this category will be test manually be the individuals having some knowledge of that domain. We have used 4GB RAM, Sesame database to store the RDF data, Netbeans 7.1 IDE for fetching the result from the RDF storage. For preprocessing we have chosen standard stop words list.

5. Possible Usage

There are two main usages of the this approach in the semantic grouping of different sources based on their keyword and maintain a search engine based on these keywords. As indicated in the above section, there

are quite a few approaches for applying ARM on features collected from RKBExplorer. Nevertheless, the approach should be used only as a complement with the web search engines based on the semantics. Previous techniques for grouping different datasets are based on the similarity measures. We propose a machine learning approach for the keyword prediction combined with the two datasets IEEE and ACM respectively. Comparing the whole documents are time consuming and needs more computationally expensive machines. However predicting keywords and give the result of related papers based on certain keywords (directly related or their relatedness generated by machine learning algorithms) may be a good idea for get rid of former expenses.

6. Conclusions

With the help of the prediction task we provide a framework which can search after the knowledge is boosted by the prediction of Association Rule Mining algorithm. Here the feature selection problem is automatically getting cured because we have taken only relevant features that was suggested by RKBExplorer. Now these keywords are used for searching of related papers or we can also used as the category of that paper for a multi label classification task. RDF data growing rapidly in current era, most of the applications are now using this information to provide good/extra content to the user. Utilization of this structured knowledge provides the backbone to the application [5] so our future aim is to exploit the RDF data for other various domains.

Acknowledgements

We would like to convey our deepest gratitude to graduate students Ankit Bathla, Monica Singh, Anchit Gupta for their understanding and throughout hard work.

References

1. Bizer C, Heath T, Berners-Lee T, Linked Data-The Story So Far, International Journal on Semantic Web and Information Systems, 5(3): 1–22, 2009.
2. Paulheim H, Furnkranz J, Unsupervised Feature Generation from Linked Open Data, In: International Conference on Web Intelligence, Mining, and Semantics (WIMS'12), pp. 31. ACM, 2012.
3. Venkata N P K, Ryutaro I, Vyas OP, LiDDM: A Data Mining System for Linked Data, volume 813 of CEUR, LDOW, March 29, 2011.
4. Kushwaha N, Singh B, Mahule R., Vyas OP, Using Semantics of Linked Open Data Cloud for Explication of recommender System, CIIT 2013.
5. Glaser H, Millard I C, Jaffri, A, RKBExplorer.com: A Knowledge Driven Infrastructure for Linked Data Providers. ESWC, pp. 797-801. Springer, 2011.
6. Q Zhao, S. Bhowmick, Association Rule Mining: A Survey. Technical Report, CAIS, Nanyang Technological University, Singapore, 2003.
7. Agrawal R, Srikant R, Fast Algorithms for Mining Association Rules. Proceedings of the 20th VLDB Conference, Santiago, Chile, pp. 487-499, 1994.
8. Antoniou G, Harmelen F, A Semantic Web Primer. Second Edition, The MIT Press, London, England, 2008.
9. Ducharme B, Learning SPARQL, O'Reilly, Gravenstein Highway North, Sebastopol, 2011.
10. Yu L, A Developer's guide to the Semantic Web. Springer, 2011.