LEVERAGING BIBLIOGRAPHIC RDF DATA FOR KEYWORD PREDICTION WITH ASSOCIATION RULE MINING (ARM)[1]

Nidhi Kushwaha* *and O P Vyas*

Department of Information Technology, Software Engineering Lab, Indian Institute of Information Technology, Allahabad, India
Emails: kushwaha.nidhi12@gmail.com*, *opvyas@iiita.ac.in*

ABSTRACT

The Semantic Web (Web 3.0) has been proposed as an efficient way to access the increasingly large amounts of data on the internet. The Linked Open Data Cloud project at present is the major effort to implement the concepts of the Seamtic Web, addressing the problems of inhomogeneity and large data volumes. RKBExplorer is one of many repositories implementing Open Data and contains considerable bibliographic information. This paper discusses bibliographic data, an important part of cloud data. Effective searching of bibiographic datasets can be a challenge as many of the papers residing in these databases do not have sufficient or comprehensive keyword information. In these cases however, a search engine based on RKBExplorer is only able to use information to retrieve papers based on author names and title of papers without keywords. In this paper we attempt to address this problem by using the data mining algorithm Association Rule Mining (ARM) to develop keywords based on features retrieved from Resource Description Framework (RDF) data within a bibliographic citation. We have demonstrate the applicability of this method for predicting missing keywords for bibliographic entries in several typical databases.

Keywords: Keyword generation, Linked Open Data Cloud, Data mining, Query-answering system, RDF, Association rule mining

1 INTRODUCTION

Knowledge discovery is the process of identifying novel, valid, and interesting patterns and knowledge inside data collections. Utilizing background knowledge can help in discovering those patterns as well as finding completely new information by combining the original data with additional data from different sources. In recent years, Linked Open Data Cloud (LOD) has become a large source of open background knowledge. The information inside the cloud is in the form of triplets, i.e., Subject→Predicate→Object, which follow the well defined standard of the Resource Description Framework (RDF). The two main steps of leveraging this knowledge are preprocessing and developing a specialized mining method. The paper introduces a method that follows both of these steps. The authors provide a method to utilize this knowledge for predicting tags for papers in databases. To do this, the authors extract the RDF bibliographic information residing in the RKBExplorer, a part of the LOD cloud that contains bibliographic information. The rationale is that such a strategy allows the prediction of a paper's tags through the data mining algorithm, ARM. This algorithm uses features from semantic data that are present in the cloud in RDF format. This method can help to identifying tags/keywords not defined explicitly by the author or predicting new tags. This is beneficial for searching for related papers. Furthermore, our approach can be used to complement the search engine and can work in conjunction with a paper personalization wesite (http://www.bibsonomy.org/) to recommend relevant keywords/tags.

The rest of this paper is structured as follows. Section 2 introduces a background survey. Section 3 explains the theoretical framework, and Sections 4 and 5 discuss the complexity and current implementation of this framework. Section 6 introduces a performance analysis followed by possible usage of the proposed method in Section 7. We conclude with a review of current challenges and future direction in Section 8.

[1] Paper presented at 1st International Symposium on Big Data and Cloud Computing Challenges (ISBCC-2014) March 27-28, 2014. Organized by VIT University, Chennai, India. Sponsored by BRNS.

2 BACKGROUND

The Linked Open Data (LOD) Cloud project was begun in 2008 by Tim Berners-Lee (Bizer, 2009) with the goal of open source sharing of information on the web through globally connected data using unique URIs. Publishing their data on the internet gives users the opportunity to take advantage of various application domains. Google Rich snippet and Yahoo Search monkey are good examples of search engines that use Resource Description Framework (RDF) data embedded in less informative XML documents. A wide variety of applications and browsers support RDF data, thus illustrating their continuous growth and utility within the current working environment. The traditional web consisted of millions of interconnected pages. The logic behind this connectivity, however, was missing, causing problems for future users.

Ontology development needs specialized personnel with good knowledge of the field of interest. Because of domain dependence, the ontologies that have been developed have different types of connections and concept names (Yu, 2011), leading to the problem of how to combine them. One solution put forward by ontology engineers is the use of ontology development tools and the important predicate link known as "owl:sameAS". The Linked Open Data Cloud,which is continuously growing, did this work under norms defined in Bizer (2009) and Yu (2011). The cloud has various types of domain information, including a cross domain giant "DBpedia" (Bizer, 2009). The cloud itself presents an example of diverse information sources, and much work has been done to connect this diverse information. For example, a person related by the FOAF (Friend of a Friend: http://www.foaf-project.org/) ontology may also connect with a DBLP (a computer science bibliography: http://datahub.io/dataset/rkb-explorer-dblp) ontology with a paperID by has-author relationship. The co-author of the paper might teach in the same university (University Ontology) as a student presenting this paper in a conference (Event Ontology) held in Japan (Geoname Ontology). This example illustrates the connectivity of one person to another person, to a paper, to an organization, and to a country.

Accessing all this information automatically without moving through hyperlinks was the initial idea of the LOD Cloud. In the last decades, many organizations have provided successful open source knowledge bases (Bizer, 2009). This knowledge has been utilized in many applications to give meaningful results by combining different data sources, which was only possible because they had the same structured format (Kushwaha, 2013). The RDF triplet contains information about concepts in the form of their relationship with all other related concepts. These links also have some special characteristics. Different links can attach to different objects or values in the same way that an actor in a movie can also be the director.

This is not the case, however, when we talk about bibliographic databases. According to Glaser (2011), the RKBExplorer contains information about bibliographic data that can be used to generate keywords (also known as tags) for these data. It also provides a unified view of heterogeneous data sources. The ReSIST (Resilience for Survivability in IST) project (Glaser, 2011) proposes a semantically-enabled knowledge structure. The project's aim is to provide services from different but related data sources. It aims to show the semantic knowledge from different bibliographic sources as well as related organizations and people. However, this project does not utilize the available information for predicting new knowledge that would help in a more related search.

In the next sections we describe the proposed methodology for our work and the implementation and results of the framework, and conclude in the last section with future work.

3 TRIPLET EXTRACTION THROUGH MULTIPLE RDF DATASETS

Our aim is to predict tags/keywords by using the available background knowledge to deal with the previous issue. With this approach, a user is able to search for additional relevant papers by entering just certain strings or keywords. Figure 1 shows the flow diagram of the keyword prediction method. In the first step, the datasets are downloaded from datahub and stored in the local RDF repository (for more information see Section 6). Then the direct and complementary links are extracted from each bibliographic dataset using SPARQL query language (Ducharme, 2011). How these links are extracted is explained below. In the next step, information from these links is extracted and the removal of stop words is done by applying data mining (ARM). In the ARM process, PaperID will act as the Transaction ID, and the values of the associated features act as different items related to the first transaction in a typical ARM minning algorithm (Agarwal, 1994). For testing, different data subsets are formed (called "Test Query Datasets"), which are used for performance analysis.

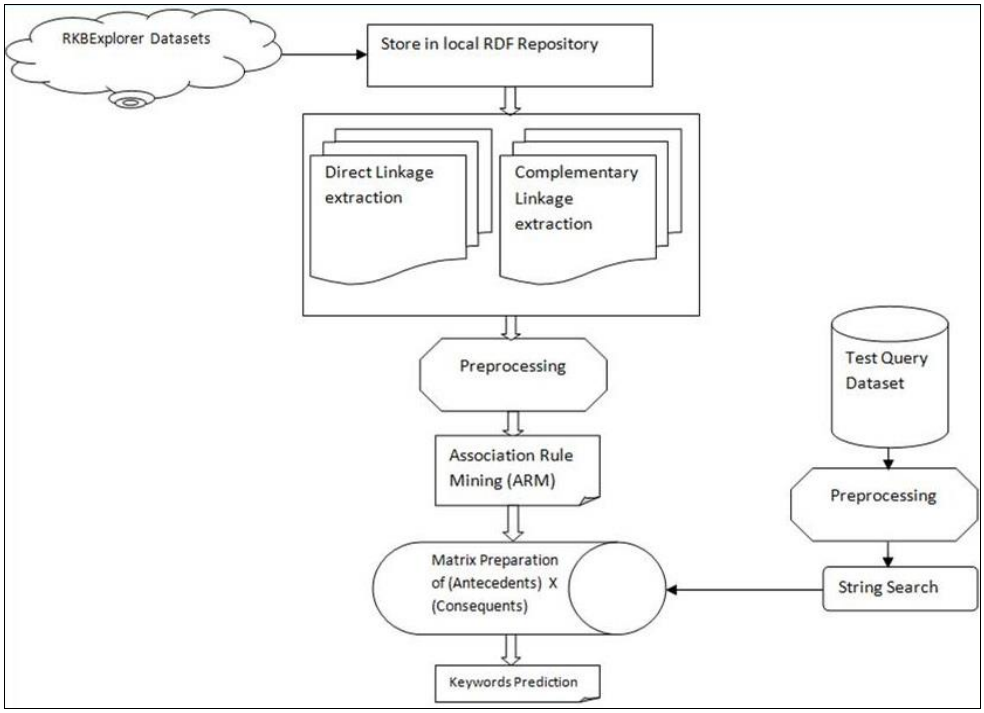

Figure 1. Framework of keyword (tag) extraction from bibliographic RDF data

We studied three bibliographic datasets

- DBLP computer science database (http://datahub.io/dataset/rkb-explorer-dblp)
- IEEE database (http://datahub.io/dataset/rkb-explorer-ieee)
- ACM database (http://datahub.io/dataset/rkb-explorer-acm)

These are very basic and are three of the most utilized datasets in bibliographic searches. Those who do bibliographic searches are not ordinary users but have enough training to be able to work with the results of the search queries (Ducharme, 2011). The amount of time used for these searches, however, is a large problem. The LOD Cloud (Bizer, 2009) is a good example of semantic connectivity among large knowledge bases. The information provided by the datasets in this cloud is semantically linked. We can consider the datasets as a huge graph in which the vertices are the subjects and objects. The linkage information is used as a predicate between the subject and the object. Therefore, the information is called a triplet. The linkage information gives us an opportunity to specifically utilize the objects or values. The authors (Kushwaha et al., 2013) have listed some interesting information about these three RDF datasets, thanks to the bibliographic RDF converter organization that provides a common ontology for all three of them (see Figures 2 and 3).

Figures 2 and 3 are excerpts of RDF graphs for two bibliographic ontologys, namely the IEEE and ACM datasets. In Figure 2, the Paper ID is represented by "http://ieee.rkbexplorer.com/id/publication-04223221" URI, which is further connected by four distinct features: "akt:has-title" , "akt:sub-area", " iai:is-ver-strongly-relate-to", and "akt:has-ieee-keyword". The values of these features are explained by the corresponding eclipse values. In this case, the Paper ID is the *Subject*, links are the *Predicates*, and the values of these links are called *Objects* (Similarly Figure 3 is concerned with the Paper ID "http://acm.rkbexplorer.com/id/192016" inside the ACM ontology). The links or predicates use two vocabulary ontologies (akt and iai) to describe the paper's features. Review has shown that some common information, such as "sub-area-of", "has-author", "fullname", "has-title", "has-date", "year-of", is present in all three datasets. Some of the information has a similar meaning but different predicates, such as "cites-publication-reference" in DBLP and ACM and "is-very-strongly-related-to", "is-strongly-related-to", and "is-related-to" information in IEEE. Another example of this is "has-ieee-keyword"in IEEE and "address-generic-area-of-interest" in the ACM dataset. In our discussion, we called these similar terms and complementary terms, and they are surrounded by boxes in Figure 4. The features surrounded with red boxes represent direct terms and features while the purple boxes represent complementary terms and features. Our aim is to utilize complementary terms as well as direct information from the three datasets using the fewest number of preprocessing steps for an ARM prediction of new keywords and tags.

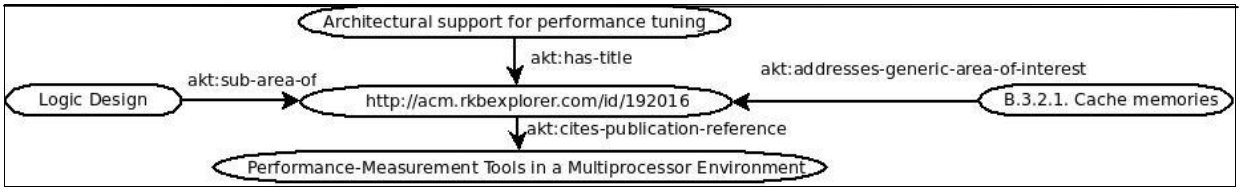

Figure 2. IEEE-RDF graph excerpt

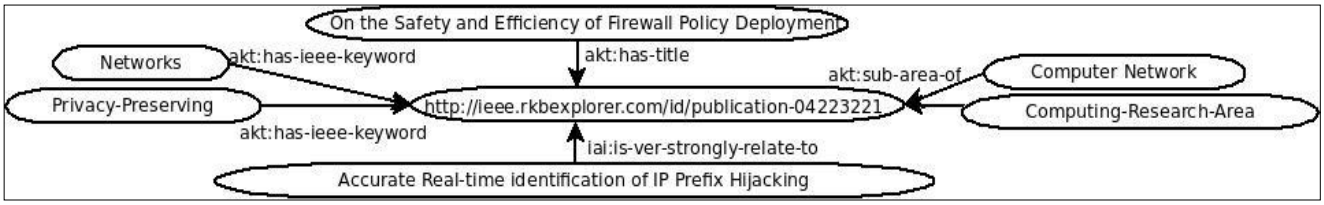

Figure 3. ACM-RDF graph excerpt

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

Figure 4. Predicates of the DBLP/IEEE/ACM datasets

4 COMPLEXITY OF THE ALGORITHM

The algorithm utilizes the information from RDF datasets to predict the unassigned and unknown keywords for publications, thereby facilitating their location by searchers. The algorithm combines two RDF datasets for prediction of keywords. This section contains the analysis of the algorithm in Figure 5. In step 1, we load the RDF graphs of the datasets (here IEEE and ACM), which are linear in nature and have a time complexity of O(1) because the processing is done on the system itself. SPARQL querying in step 2(a) also consumes linear time while step 2(c) checks the predicate for direct and indirect linkage that requires O(m*n), where 'm' is the total number of predicates belonging to different datasets and 'n' is the number of RKBExplorer datasets being considered.

The normalization step depends on the number of objects that belong to these direct and indirect links. We consider 'h' as the average number of objects for a particular predicate link. For example, direct property 'akt:has-title' always has one object while indirect property 'akt:sub-area-of' can have more than one object. Therefore, the worst case time complexity for normalization will be O(m*h). Due to appending each object that belongs to a specific subject or paper ID, step 4 will have the same complexity as the previous step. Moreover,

the a priori step takes three inputs: the normalized datasets, the support, and the confidence value. We have taken support 20% and also checked it at 40%, keeping confidence value (80%) constant. We assume 'c' number of candidate sets are generated during processing. Here, candidate set refers to the frequent terms that appear in the data. The ARM approach will be similar to a typical case (Agarwal,1994), except in the place of TransactionID we consider PaperID and in the place of different items related to one transactionID, we consider different features of the corresponding PaperID. These features are fetched from the RDF datasets as explained in Section 3. The algorithm has a O(m*h*c) running time complexity. The overall time complexity for the proposed algorithm is O(m*h*c), in our case, because h>n and c>1. This means the cost of the generating the algorithm depends on three factors: the number of predicates in the different datasets, the average number of objects, and the number of candidate sets generated during the ARM 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 (Broekstra, 2002):

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

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

c) If $p_i=p_j=p_k$ 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, $p_i \in D_i$, $p_j \in D_j$, $p_k \in D_k$.. For given table.1 D_{dblp} , D_{ieee} , D_{acm}..

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

If ($O_i \in$ Str) then,

{

If (O_i==l_i), remove the term

else If ($O_i \in$ 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 'norO_i' related to Direct & Indirect, by below steps, for a particular subject S_i,

If ($norO_i \in S_i$) then, append O_{ii}, O_{ij}, O_{ik} in one row that belongs to one instance in dataset. Similarly prepare list for all the $S \in D_i$, D_j , D_k & called it 'Đ'.

5. A priori (Đ, Supp, Conf)

For p=1 to h

For q=1 to m

{

M_{pq} =Conf(p,q)

}

Return M

For Testing:

Steps for obtaining test results:

T={t_1,t_2,t_3....t_n} a set of test queries, where t_i(w_1,w_2,w_3,...w_n) and $t_i \neq \emptyset$ after normalization.

If (w_1,w_2,w_3,...w_n) match with M_p, where $W_i \in t_i$ 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'.

Figure 5. Proposed algorithm for keyword prediction of new keywords and tags from RDF datasets

5 IMPLEMENTATION OF DETAILS FOR RDF STORAGE, EXTRACTION, AND MINING

We used approximately 300 pieces of data from IEEE and 500 pieces of data from ACM. Because of inadequate knowledge of the keywords, we were unable to consider the DBLP data. After the direct and complementary features were selected, the preprocessing step removed all the stop words and formated the keywords as discussed above (see Figure 6). At this point the dataset was ready for the ARM generation. (Zhao, 2003; Agarwal, 1994) (see Figure 7)**.** Next, we inserted the results into the matrix p*q, where 'p' is the antecedent and 'q' is the consequent in the generated rule (see Table 1)**.** Then we tested our rules with the preprocessing step, where we removed all the stop words present in the datasets and used these tokens as a string search (see Figure 8). The strings were then compared to the antecedents of the matrix, and the corresponding consequents were the results. These selected results were prioritized. The topmost result had a confidence greater than all the others in the corresponding vector of matched strings. For example, in Table 1, if the search string is matched with 'network', the order of recommendation is 'cryptography', 'data', and 'information'. In Figure 9, the test query is the title of the paper whose keyword has not been previously assigned.

algebra security systems_analysis security_of_data Boolean_algebra
Concerning modeling computer security systems_analysis security_of_data
Security specifications security_of_data specification_languages
Extended discretionary access controls distributed_processing operating_systems_(computers) security_of_data
fault tolerance approach computer viruses fault_tolerant_computing security_of_data programming
prototype real-time intrusion-detection expert system real-time_systems security_of_data supervisory_programs expert_systems
Views security objects multilevel secure relational database management system data_structures relational_databases security_of_data
ASDViews [relational databases] data_structures relational_databases security_of_data query_languages
Inference aggregation detection database management systems relational_databases knowledge_engineering query_languages

Figure 6. Result after normalization step

{key} => {cryptography} support: 0.02120141342756184 confidence: 0.5
{model} => {security} support: 0.02120141342756184 confidence: 0.35294117647058826
{systems} => {secure} support: 0.02120141342756184 confidence: 0.24000000000000002
{secure} => {systems} support: 0.02120141342756184 confidence: 0.3
{analysis} => {security_of_data} support: 0.038869257950530034 confidence: 0.5789473684210525
{computer_network_management} => {authorisation} support: 0.024734982332155476 confidence: 0.4666666666666666
{Using} => {security_of_data} support: 0.02120141342756184 confidence: 0.8571428571428572
{key} => {authorisation} support: 0.02120141342756184 confidence: 0.5

Figure 7. Rule generation with ARM

```
On a Modeling Framework for the Analysis of Interdependencies in Electric Power Systems
enter threshold
10
recommended keywords
security of data
security of data
enter book, article or paper name
Graphical aids to constructing parallel programs summary
```

Figure 8. Search query results

	Leap-frog packet linking and diverse key distributions for improved integrity in network broadcasts	*An adaptive distributed system-level diagnosis algorithm and its implementation*
1	Cryptography	Distributed Processing
2	Authorisation	System
3	Data Integrity	Algorithm
4	Security of data	Adaptive
5	Network	Data

Figure 9. Top-5 recommendation for search queries

Table 1. p*q matrix example

Antecedents/Consequents	*information*	*data*	*cryptography*
network	0.005	0.06	0.78
integrity	0.30	0.68	0.17
security	0.45	0.62	0.80
algorithm	0.23	0.23	0.74

6 PERFORMANCE EVALUATION

To evaluate the proposed method, we have divided the whole dataset, composed of RKBExplorer information from different publishing groups such as ACM and IEEE, into 75% training data and 25% test data. For our purposes, we used the information from a total of 800 papers for model training. To make the results more effective, we first prepared a dataset dedicated to a specific domain, in our case "networking" and "information retrieval."

Next, we prepared matrix M_{pq}, from which we fetched the results after pre-processing the test query. After obtaining the results, we matched them with known keywords already associated with particular papers. The precision calculation was done by matching the known and predicted values. Figure 10 shows those results.

With increasing support values, the precision increased. For evaluation we also introduced one additional feature that judged the effectiveness of the output using manual expertise, called 'Novelty'. This found new keywords that were not previously attached to the specific paper but were determined by our model. These novel keywords were generated by considering the papers that occurred in one dataset that might not have been present in another publishing group. Our process is able to obtain results from other keywords found in other publishing groups. The newly generated keywords in this category were test manually by individuals having knowledge of that particular domain. We used a 4GB RAM, Sesame database to store the RDF data and net bean 7.1 IDE for fetching the results from RDF storage. For pre-processing we chose a list of standard stop words available from http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words.

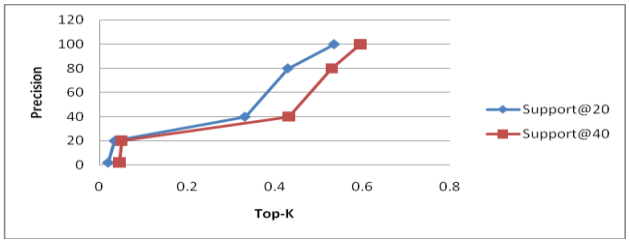

Figure 10. Accuracy measurement

7 POSSIBLE USAGE

There are two main uses for our approach: the semantic grouping of different sources based on their keywords and improvement of search engine performance. As indicated in Section 2 in the discussion of the ReSIST project (Glaser ,2011), there are quite a few approaches that apply ARM to features collected from RKBExplorer. Nevertheless, our approach can be used as a complement to the web search engines based on semantics. Previous techniques for grouping different datasets were based on similarity measures. We have proposed a machine learning approach for keyword prediction and demonstrated its effectiveness using two datasets, IEEE and ACM. Comparing whole documents is time consuming and needs computationally expensive machines. However, predicting keywords and finding related papers based on certain keywords (directly related or their relatedness generated by machine learning algorithms) will considerably lessen the expense of the search. This approach combines the social and semantic webs by using semantic information from keywords and tags gathered by the social media. The website based on Bibsonomy data (http://www.bibsonomy.org/) is a good example of a paper tag or keyword recommendation.

8 CONCLUSION

In this paper we explain a new approach for keyword prediction utilizing bibliographic RDF datasets. The main goal of this approach is to convert RDF data into a form in which a data mining algorithm can be easily used. With the help of the prediction task, we provide a framework that searches after knowledge is boosted by predictions of the Association Rule Mining algorithm. Here the feature selection problem is automatically fixed because we have taken only relevant features suggested by RKBExplorer. Once these tags or keywords are predicted using the algorithm, some of those (above some support threshold) will be associated with the paper or the subject itself. Then these keywords are used to search related papers, or they can also be used as the paper's category for a multi label classification task. As the RDF data grow rapidly in the current era, most applications now use this information to provide good/extra content to the user. Utilization of this structured knowledge provides a backbone for the application (Glaser, 2011); therefore our aim for the future is to exploit the RDF data for other various domains using different efficient data mining techniques.

9 ACKNOWLEDGEMENTS

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

10 REFERENCES

Agarwal, R. & Srikant, R. (1994) Fast Algorithms for Mining Association Rules. *Proceedings of the 20th VLDB Conference, Chile*, pp 487-449.

Bizer, C., Heath, T., & Berners-Lee, T. (2009) Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems 5*(3), pp 1-22.

Broekstra, J. & Harmelen, A.K. (2002) Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *First International Semantic Web Conference (ISWC) 2342*, Springer: Sardinia, Italia, pp 54-68.

Ducharme, B. (2011) *Learning SPARQL*, Sebastopol: Gravenstein Highway North.

Glaser, H., Millard, I.C., & Jaffri, A. (2011) RKBExplorer.com: A Knowledge Driven Infrastructure for Linked Data Providers. *ESWC*, Greece: Springer, pp 797-801.

Kushwaha, N.S., Singh, B., Mahule, R., & Vyas, O.P. (2013) Using Semantics of Linked Open Data Cloud for Explication of Recommender System. *Proceedings on ComNet CIIT & ITC*, Elsevier, pp 357-364.

Paulheim, H. & Furnkranz, J. (2012) Unsupervised Feature Generation from Linked Open Data. *International Conference on Web Intelligence, Mining and Semantics (WIMS'12)*, New York, USA: ACM.

Venkata, N.P.K., Ryutaro, I., &Vyas, O.P. (2011) LiDDM: A Data Mining System for Linked Data. Workshop on Linked Data on the Web. CEUR Workshop proceedings.

Yu, L. (2011) *A Developer's Guide to the Semantic Web*, Springer: Berlin.

Zhao, Q. & Bhowmick, S.S. (2003) *Association Rule Mining: A Survey*, Singapore:CAIS.

(Article history: Received 1 July 2014, Accepted 19 September, Available online 6 October 2014)