International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-6)

Research Article

June 2015

Semantic Web-based Recommendation: Experimental Results and Test Cases

S. Manoj Kumar
M. Tech, School of IT
JNTUH, Hyderabad, India

K. Anusha
M. Tech, School of IT
JNTUH, Hyderabad, India

K. Santhi Sree
Professor in School of IT
JNTUH, Hyderabad, India

Abstract—

Web contents including multimedia contents, Web services and documents, Web personalization recommendation plays an important role to meet the users' information needs on the Web. Most of these commercial recommender systems are implemented based on Collaborative Filtering (CF), which is the method of considering preferences of multiple users in a collective way. CF assumes that if two users have shown similar interest on the same set of contents, in future also they may show a similar interest-pattern in choosing contents. However, users who have certain tastes on one specific category of contents, they may behave differently in choosing contents from other categories. Moreover, this collaborative filtering approaches may not work efficiently in certain situations like sparse data sets (data sparsity), where there are a small number of contents or a limited number of users in the content categories. Firstly, we proposed a framework of utilizing Linked Data to expand keywords extracted from content metadata to semantically enriched information. Secondly, we proposed a way of comparing semantic clusters generated from users' viewing history to identify similar interests of users. We collect more data about users' viewing history as well as the metadata about more number of contents. Especially, viewing history datasets for a longer period of time is essential to demonstrate the practicality and effectiveness of our approach in terms of recommending more personalized contents. In addition, we will integrate users' social network data to identify their implicit interests. We expect that it will be possible to produce more personalized semantic clusters from users' social network data, and to improve the users' satisfaction on the recommended contents that reflect more personalized preferences of the users. We proposed an approach of recommending contents across different categories by taking into consideration of semantic information of content and user interests. To find the appropriate semantics of the content extracted from user's viewing history we use the Linked Data as the source. Then based on that similarity of semantics and relevance of the content we group together into semantic clusters. Our approach recommends the contents to the general users based on the leading user groups. The leading user groups are the group of users who frequently consume contents.

Keywords— Semantic Cluster; Linked Data; Recommender System; RDF; Sparql;

I. INTRODUCTION

A. Clustering

Clustering is a data mining technique that categories the data into multiple groups, called as clusters. The main property of data clustering is inter cluster similarity has to be maximized and intra-cluster similarity has to be minimized. All the patterns that lie in one cluster are similar to one another and dissimilar when compared to clusters in the other clusters .i.e. the distance between the patterns that lie in one cluster is less and similarity between patterns that lie in two different clusters is more. The various types of clustering techniques are hierarchical algorithms, partitioning algorithms, grid based, density- based and model- based algorithms. K-means and K-mediods are partitioning algorithms , Agglomerative and Divisive are Hierarchical clustering techniques.

B. Recommender Systems (RS)

Recommender systems are more popular in commercial as well as in the research fields, where lot of techniques have been suggested for providing recommendations. Recommender systems are part of our day to day life applications; they recommend very large (bulk) collections of items. These systems are in the every domain i.e. songs, movies, books, etc. Example for recommender systems are Amazon, Pandora, etc. These systems (RS) gives the list of recommended items to the user that he prefer. It also gives the recommendations by predicting how much he prefer each item. These systems may help users to decide on suitable items, and ease the task of finding preferred (appropriate) items in the collection. There are different kinds of recommender

C. Semantic web

The Semantic Web [2] is a web of data. This semantic web is connected completely i.e. in the form of complete graph, which can be easily readable by machines. With the invention of the internet we can find lot of data on the web every day. There is lot of data we all use every day, and it is not part of the web. Example you can see your stipend details, photographs in the internet and you can see your timetable in the calendar. But can you see your photographs in a calendar to see what you were doing when you took them? Can you see your stipend details in a calendar? Because lack

of a web of data. The main reason is that data is controlled by applications individually, and each application keeps data it to itself.

The Semantic Web deals with the two major things one is the common formats to combine (integrate) the data from the different sites. The web mainly focuses on the interchanging of the data in the form of documents instead of the common formats. The second one is language in the real word there are different languages the data is exchanging. The semantic web allows the human or a machine to find the objects from the different sources i.e. it starts from one database and then moving to the unending data sets.

D. Linked Data

The Semantic Web is a Web of Data, to make the Web of Data a reality, it is very important to have the huge amount of data on the Web available in a standard format(XML, RDF, etc.), reachable and manageable by Semantic Web tools. It is not only does the Semantic Web need access to data, but the other main feature is relationships *among data* should be made available. This collection of interrelated datasets on the Web can also be referred to as Linked Data. [14] Linked Data is not a specification, but it is a set of best practices for providing a data infrastructure that makes it easier to share data across the web. The common format used to create the linked data is the RDF.
Examples of linked data are DBPedia, Freebase, etc,.

E. RDF

RDF [13] is the short form for Resource Description Framework. It is a frame work which describes the resources in the web. The main motto of RDF is that reading and understanding by computers easily. It provides a means to publish machine-readable and human-readable vocabularies to encourage the reusing of metadata semantics from different communities. It is a data model which is similar to classical conceptual modeling approaches such as E-R Model, as it is based upon the idea of making statements about resources in the form of Subject–Predicate–Object (RDF) expressions are also known as triples in RDF terminology. The subject denotes the resource, and the predicate denotes aspects of the resource and expresses a relationship between the subject and the object. To remove any ambiguity from the information stated by a given triple, the triple's subject and predicate must be URIs.

For example, "The T-Shirt has the color White" in RDF is as the triple: a subject denoting "the T-Shirt", a predicate denoting "has", and an object denoting "the color White".

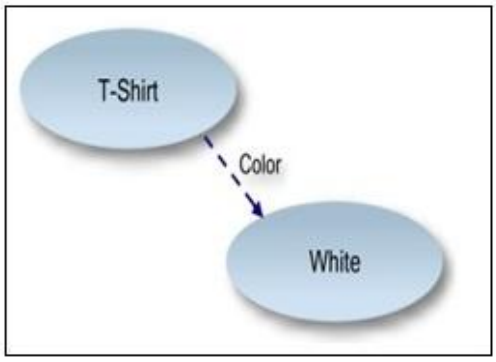

Fig 1. T-Shirt has the color White

F. sparql

SPARQL[11] is the short form of SPARQL Protocol and RDF Query Language. It is an RDF query language, i.e., a semantic query language for databases. It can retrieve and manipulate data stored in Resource Description Framework (RDF) format. It is a machine friendly protocol it is not able to interpret by humans, so that it requires an interface i.e. known as endpoint. This endpoint allows the user to enter the query and display in the human understandable format. These endpoints use the HTTP Protocol.SPARQL allows users to write queries against data that can loosely be called "key-value" data that follows the RDF. The entire database is thus a set of "subject-predicate-object" triples. It contains capabilities for querying required and optional graph patterns.

Example:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
   ?person foaf:name ?name .
}
```

Result:

Name
"mc schrefel"
"John Klensin"
"Libby Miller"
"Henrik Nielsen"
"John Markoff"
"Edd Dumbill"

II. RELATED WORK

In this section, we review the state-of-the-art recommendation systems that use collaborative filtering as the method to improve the quality of recommendation. *Collaborative filtering* is usually divided into two types: memory-based and model-based approaches [7]. The memory based CF algorithm checks the similarity between users or between items based on a collection of user preferences for the target items. The memory-based approaches rely mostly on the existing rating history rather than using external information about the users or items. Therefore, the recommendation result may be unreliable if the rating records do not exist. In other words, there are the cold start and data sparsity problems in the memory-based approaches. Model-based CF recommends items by using a rating model that is built based on training datasets. Usually, the model-based CF shows better performance than the memory based CF because it uses probabilistic approaches. However, the main drawbacks of this approach are the computational complexity of the model-building process and information loss during dimensionality reduction. Recently, many studies have been conducted to improve the matrix factorization method to predict user rating in a more precise way [9].

SoRec [4] proposes a recommender system that uses matrix factorization with considering social network information. The intuition in this approach is that a user's social network may affect the user's behavior of consuming items. In this approach, a high-weight value is assigned to the trustful relations between users in the social network. Although this approach can deal with the data sparsity and cold start problems, it does not consider domain-specific user preferences that are essential to recommend items effectively across different domains. Users who have similar preferences in one domain may show different preferences in other domains. Recent studies [1, 5, 8] propose the systems that deal with this problem effectively by using the matrix factorization method. *MCoC* [1] proposes a multiclass co-clustering method to find meaningful user-item subgroups. Subgroup information is utilized by applying collaborative filtering to cluster users and items in similar context groups. However, this approach requires the external contextual information of like-minded groups.

HeteroMF [5] extends collective matrix factorization to develop a recommender system for heterogeneous contextual domains. They introduce general latent factors and context dependent latent factors to recommend entities of one type to those of others. However, the context domain of each entity should be known beforehand. To compliment this problem, *TopRec* [8] proposes a community topic-mining recommender system. They firstly discover active users in topic groups by representing the users as the probability distribution over topics. Then, it supplies experts' rating information as the prior knowledge of the probabilistic model for topic alignment. Furthermore, it includes social network information in the topic model to consider the similarity between users. After that, it applies the probabilistic matrix factorization in different domains by considering their domain interests. Selecting the active user of each domain could increase the efficiency of clustering users in a specific domain. However, this study only uses items that are in predefined domains.

III. PROPOSED WORK

First divide the total users into Leading users and general users from the user's viewing history based on the contents consumed. The users who consume various contents intensively during a certain period of time are known as the leading users and others are known as the regular users. To divide the users we count the number of different contents that they consumed during that period.

Then send the content which is collected from the leading user to morpheme analyzer to get the phrases we call them as keywords. With this keyword we are going to access the linked data. Extract the rdf content by querying with the sparql. With these rdf content we frame the skos concept. Each concept having the data which is broader, narrower, exact match, close match than the given keyword. The more similar concepts are going to group together known as semantic cluster. Again based on the similarity we are going to group they together, called as leading user groups.

Then classify the regular users among those leading user groups. A regular user may be classified under multiple groups based on the similarity. Then figure out the different content consumed by the regular and leading user and recommend those content to the regular user.

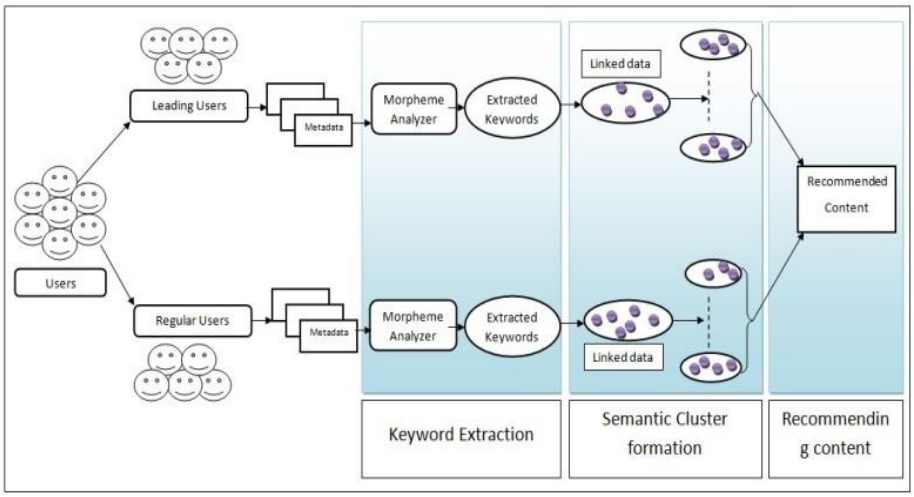

Fig 2. Overall Process for proposed Approach

A. Keyword Extraction

As the first step extract the keywords from the content consumed by the users. To extract the keywords we use the Morpheme analyzer, it is a toll to divide the sentence into the nouns and noun phrases.

B. Fetching Linked data and cluster formation

After extracting the keyword we query with each noun and noun phrases to linked data to extract the keywords. In general the linked data is in the form RDF triple. RDF triple describes the concepts, containing the keyword as the one of their property value such as label, name, or title. Each dataset in the linked data uses the different kinds of predicates, those are rdf: label, skos: lable, skos: prflabel are used in the dataset. By using these predicates we querying with the SPARQL [11] to fetch the concepts from RDF triples. [13] Here we get the SKOS [12] data as the output, and we are going to form the skos concept with skos properties such as broader, narrower, exact match, pref label, owl: sameas. By retrieving the skos data we are going to group the relevant properties into concept.

After retrieving concepts from the linked data with the keywords found in the content, we group the semantically relevant concepts together as semantic clusters. The purpose of the semantic cluster is to remove the irrelevant concepts. To find the semantically relevant concepts we use a model called concept analysis model.

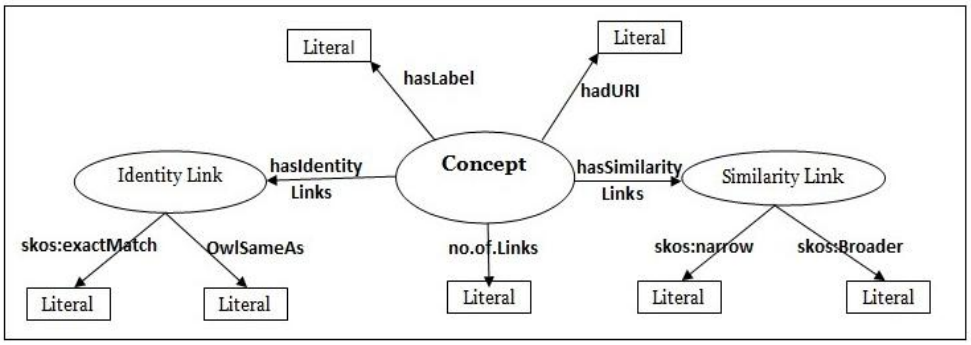

Fig 3. Concept Analysis Model

In this model each concept has a label for presentation and uri for reference. Each concept have the different properties like skos:exactMatch, owl: sameAs, skos:broader, skos:narrower, skos:closeMatch, skos:prefLabel, skos: altlabel. All these properties are connected to the concept through hasLabel, haduri. Then find the concept with the hasLabel property and get the similar concept links, like exact match, close match concept links. The concepts which are similar can be identified with skos:broader and narrower. Based on these properties we are going to represent the hierarchies of concepts. Based on the similarity we form the clusters, they are called as semantic clusters.

The basic idea of this method is to measure the relevance of the matching between a particular concept the user is interested in and a concept describing the item. [6] (In Fig 3, two examples are shown, in which the user's interest is the parent of the item concept.) We can distinguish two types of matching:

The item concept is one of the user's interests, so the matching is perfect and the similarity is maximum.

An ancestor of the item concept (e.g., the direct parent) is one of the user's interests. In this case the similarity is calculated using the following recursive function whose result is always a real number (lower than 1).

- SIMn = SIMn-1 – K * SIMn-1* n (partial match, n>0)
- SIM0 = 1 (perfect match, n=0)

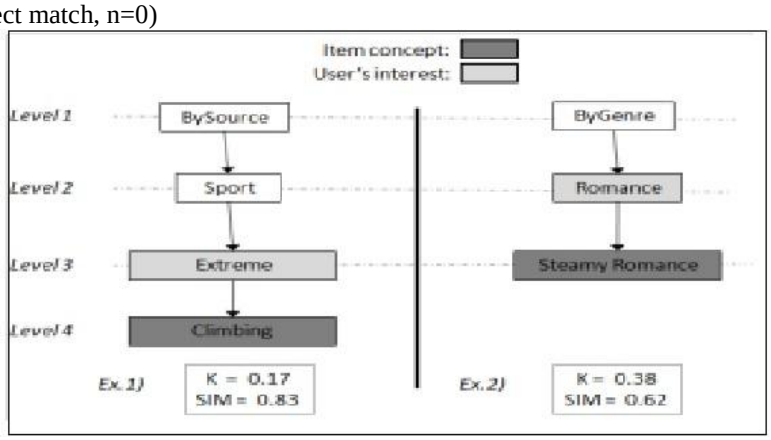

Fig 4. Similarity working Process

Where:

- n is the distance between the item concept and the user's interest (e.g., when it is the direct parent, n = 1);
- K is the factor that marks the rate at which the similarity decreases (the higher n, the higher the decrement). This factor is calculated taking into account the depth of the item concept in the hierarchy and is based on the assumption that semantic differences among upper-level concepts are bigger than those among lower-level concepts.

Algorithm
For clustering K-means [3] clustering technique is implemented to the project.
*Algorithm***:** K-means.

The *k*-means algorithm for partitioning, where each cluster's
Center is represented by the mean value of the objects in the cluster.
Input:

k: the number of clusters,
D: a data set containing *n* objects.

Output: A set of *k* clusters.
Method:

1. arbitrarily choose *k* objects from *D* as the initial cluster centers;
2. repeat
3. (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
4. update the cluster means, i.e., calculate the mean value of the objects for each cluster;
5. until no change;

C. Recommending Contents

With the help of these clusters we are now measure the similarity among the concept with the Euclidean measure. Then group the similar concepts based on the content consumption, we call them as leading user groups. Then each regular user is categorized among those groups. Then compare the content consumed by the user which is differ from the group then those content is going to make a list.

Pearson Correlation

Pearson correlation is a measure to find the similarity among the two sets of data. The correlation is in the range of -1 to 1. If the two sets x and y having 1 the we call it as positively correlated, -1 means negatively correlated, and 0 mean not at all correlated the two sets.

$$corr(x,y) = \frac{\text{co variance(x, y)}}{\text{standard deviation(x)X standard deviation(y)}}$$

$$corr(x) = \frac{\text{n}(\sum \text{xy}) - (\sum \text{x})(\sum \text{y})}{\sqrt{[\text{n}\sum x^2 - (\sum x)^2][\text{n}\sum y^2 - (\sum y)^2]}}$$

Then list of contents consumed by the leading user group and not consumed by the regular user will be recommended to the regular user.

IV. EXPERIMENTAL EVALUATION

A. MovieLens Data Set: Description

In this project we use the MovieLens dataset [10]. This data set consists of:

a) 100,000 ratings (1-5) from 943 users on 1682 movies.
b) Each user has rated at least 20 movies.
c) Simple demographic info for the users (age, gender, occupation, zip).

This file contain the following files

a) u.data -- The full u data set, contains 100000 ratings by 943 users on 1682 items.
b) u.info -- The number of items, users, and ratings in the u data set.
c) u.item -- Information about the items (movies); this is a tab separated list of movie id | movie title | release date | video release date
d) u.genre -- A list of the genres.
e) u.user -- Demographic information about the users; this is a tabseparated list of user id | age | gender | occupation | zip code.

B. Experimental steps

Sample data: For the experimental purpose we take 20 synthetic records from the movie lense. Each record contains the use_id and movie_id, movie_title, rating given by the user to that movie.

TABLE I : SAMPLE DATA OF USERS AND MOVIES

user_id	item_id	rating	item_title	genere
1	5	5	Hideous	horror(c)
1	7	4	Annie Hall	comedy(h)
1	8	5	The Tourist	romance(t)
2	5	3	Hideous	horror(c)

2	8	2	The Tourist	romance(t)
3	7	4	Annie Hall	comedy(h)
3	6	3	In Bruges	thriller(c)
3	9	3	Copycat	thriller(cr)
3	4	3	The Holiday	romance(c)
4	3	4	Pretty	comedy(r)
5	4	5	The Holiday	romance(c)
5	8	5	The Tourist	romance(t)
6	2	4	Vampira	horror(c)
7	2	4	Vampira	horror(c)
7	3	3	Pretty	comedy(r)
7	5	5	Hideous	horror(c)
8	8	4	The Tourist	romance(t)
9	5	3	Hideous	horror(c)
9	8	5	The Tourist	romance(t)
10	1	5	The Call	crime(t)

The leading users and regular users from the above table are listed below. The classification between the leading users and regular users are based on the content consumed by the users.

TABLE II : LEADING USERS

user id	count(user id) 1
3	4
7	3
1	3

TABLE III : REGULAR USERS

user id	count(user id) 1
4	1
6	1
8	1
10	1
5	2
9	2
2	2

Then get the contents consumed by the leading user to the morpheme analyzer to get the keywords.
['Hideous', 'Annie', ' Hall', 'The', ' Tourist', 'In', ' Br', 'u', 'g', 'es', 'InBurges', 'Copycat', 'The', ' Holiday', 'Pretty', ' Woman', 'Vamp', 'i', 'r', 'a', 'The', ' Call']
Then send the keywords to the linked data to get the skos concepts. We get the following concepts

1. Comedy (horror, thriller, romance)
2. Horror (comedy, romance)
3. Thriller (comedy, romance)
4. Romance (comedy, thriller)
5. Crime (thriller)

Among the skos concepts group the similar concepts by calculating the similarity among the concepts these process illustrated in the following figure (fig 4).

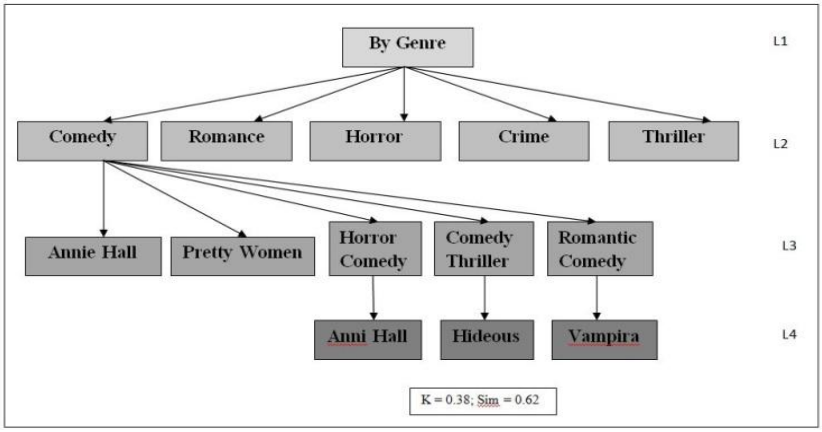

Fig 4. Similarity among the concepts

After grouping the similar concepts we perform the clustering among the semantic groups.
The clusters formed are

TABLE 4 : USER CLUSTERS

Cluster name	Users ids
Cluster-0	2, 3, 4, 6, 7, 9
Cluster-1	5, 7, 10
Cluster-2	1, 3, 5, 8, 9

Based on the clusters formed we are classify the regular users under the leading user clusters and then we recommend the contents that are consumed by the leading user and not consumed by the regular user.
Example:
Suppose recommend the contents to the user-4 are:
The user-4 is classified under the user groups are 2, 3, 6, 7, and 9. The content not consumed by the user-4 are recommended contents are as follows:

1. The Tourist (2010), 5.0
2. The Holiday (2006), 4.912
3. Vampire (1974), 4.626
4. Annie Hall (1977), 4.479
5. Hideous (1997), 4.388
6. In Bruges (2008), 4.335

V. CONCLUSION

In this paper, we extended an approach of recommending interesting contents to users with the help of semantic clusters. The semantic clusters can be generated from user's viewing history. The proposed approach recommends contents by semantic similarity between leading user groups and general users. Firstly, we extract the keywords with help of morpheme analyzer from content metadata. Then we utilized the linked data to expand the keywords. With help of linked data we extracted the relevant concepts which are associated with the relevant keyword. Secondly we identify the similar interests of users by comparing generated semantic clusters of users viewing history. In our future work we collect more data to show the effectiveness of our approach. In addition we will integrate the users social information, i.e. social network related content to identity the user's personal interests. There is a possibility to produce personalized semantic clusters from user's social network.

ACKNOWLEDGMENT

The authors express their deepest gratitude to Dr. A. Govardhan, Director, and Dr. K. Santhi Sree, School of Information Technology, Jawaharlal Nehru Technological University Hyderabad, for their support and encouragement.

REFERENCES

[1] B. Xu, J. Bu, C. Chen, and D. Cai, "An exploration of improving collaborative recommender systems via user-item subgroups," in Proceedings of the 21st international conference on World Wide Web, 2012, pp. 21-30.
[2] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). *The Semantic Web*. Scientific American, 284(5):34-43.
[3] Jaiwei Han, Micheline Kamber. "*Cluster analysis, Data mining: concepts and techniques", ELSEVIER, ISBN 13:978-1-55860-901-3, pp 401-02*
[4] H. Ma, H. Yang, M. R. Lyu, and I. King, "*Sorec: social recommendation using probabilistic matrix factorization*," in Proceedings of the 17th ACM conference on Information and knowledge management, 2008, pp. 931-940.
[5] J. Mohsen and L. Lakshmanan, "*HeteroMF: recommendation in heterogeneous information networks using context dependent factor models*," in Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 643-654.
[6] Victor Codina and Luigi Ceccaroni. 2010. Taking *Advantage of Semantics in Recommendation Systems*. René Alquézar, Antonio Moreno, and Josep Aguilar (Eds.). IOS Press, Amsterdam, The Netherlands, 163-172.
[7] X. su and T. M. Khoshgoftaar, "*A survey of collaborative filtering techniques*," Advance in Artificial Intelligence, pp. 4:2-4:2, 2009.
[8] X. Zhang, J. Cheng, T. Yuan, B.Niu, and H. Lu, "*TopRec: domainspecific recommendation through community topic mining in social network*," Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 1501-1510.
[9] Y. Koren, R. Bell, and C. Volinsky, "*Matrix factorization techniques for recommender systems*," IEEE Computer, 2009, vol. 42.8, pp. 30-37.
[10] http://files.grouplens.org/datasets/movielens/ml-100k/
[11] http://www.learningsparql.com/

[12] http://www.w3.org/2004/02/skos/
[13] http://www.w3.org/RDF/
[14] W3C SWEO Community Project Linking Open Data http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/ LinkingOpenData.
[15] K Anusha, Manoj S Kumar and Santhi K Sree, *Case Study: Outlier Detection on Sequential Data*, International Journal of Computer Applications 112(8):29-35, February 2015.
[16] K Anusha, Manoj S Kumar and Santhi K Sree, *Case Study: Semantic web-based Recommendation* Journal of Innovations in Computer Science and Engineering (JICSE) Vol. 4(2). [Accepted]

