I.J. Modern Education and Computer Science, **2017, 2, 55-61**
Published Online February 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2017.02.07

Towards Semantics-Aware Recommender System: A LOD-Based Approach

Asmaa Fridi
EEDIS Laboratory, Djillali Liabes University of Sidi Bel Abbes, Sidi Bel Abbes, 22000, Algeria
Email: fridi.asmaa@univ-sba.dz

Sidi Mohamed Benslimane
LabRI-SBA Laboratory, École Supérieure en Informatique, Sidi Bel Abbes, Algeria
Email: s.benslimane@esi-sba.dz

***Abstract*—**Recommender systems have contributed to the success of personalized websites as they can automatically and efficiently select items or services adapted to the user's interest from huge datasets. However, these systems suffer of issues related to small number of evaluations; cold start system and data sparsity. Several approaches have been explored to find solutions to related issues. The advent of the Linked Open Data (LOD) initiative has spawned a wide range of open knowledge bases freely accessible on the Web. They provide a valuable source of information that can improve conventional recommender systems, if properly exploited. In this paper, we aim to demonstrate that adding semantic information from LOD enhance the effectiveness of traditional collaborative filtering. To evaluate the accuracy of the semantic approach, experiments on standard benchmark dataset was conducted. The obtained results indicate that the accuracy and quality of the recommendation are improved compared with existing approaches.

***Index Terms*—**Recommender system; Collaborative filtering; Content-based filtering; Linked Open Data; Clustering.

I. Introduction

Recommender Systems are software tools and techniques exploiting the maximum amount of data available in the net to meet user needs, minimize time spent on research, but also to suggest relevant items that would not have spontaneously and so consulted increase its overall satisfaction. The quality of recommendations is dependent on the nature of details (in terms of quality and quantity) available to users [1]. The data available on the web are in large quantities that needs a filtering to make the best data. Several techniques and algorithms have been proposed in literature for performing recommendation, including collaborative, content-based, knowledge-based and other techniques. However, these approaches suffer from several problems such as cold start, the sparsity and new user or new item. A variety of hybrid recommender systems have also been introduces in an attempt to combine two or more recommendation to produce more effective recommendations and gain better performance with fewer of the drawbacks of any individual one [2].

Thanks to the Semantic Web standards and technologies, a massive amount of RDF data have been published using open and liberal licenses. The availability of such data is for sure an opportunity to feed personalized information access tools such as recommender systems [3].

In this paper, we propose a novel approach that collects data from Linked Open Data (LOD) in order to improve the accuracy and the quality of recommendation systems. This is achieved by a pipeline of processes divided into two phases. The first phase include the enrichment of semantic description of items, their clustering and generating of data model. The second phase consist of generating, filtering and ranking paths. Finally, the Top-K items targeted by the best paths are recommended to the user.

The rest of the paper is structured as follows. In section 2, we provide an overview of related work on acquisition of semantics from unstructured and semi-structured resources. In section 3, we outline the proposed approach and discuss the detailed steps. Section 4 introduces an experimental methodology to evaluate the approach. Finally, section 5 concludes the paper and points directions for future work.

II. Related Work

Several approaches have been proposed to incorporate semantic in recommenders system, particularly LOD. In the following we review some recent literature on LOD-based approaches, that will be composed in three categories relatives to the classification.

A. Collaborative filtering

Heitmann and Hayes [4] proposed an open music system that collected data items and users of different LOD sources and turn them into an RDF representation to apply the collaborative filtering.

Yang et al. [5] introduced a new approach for enhancing Slope-One algorithm using semantic technologies. They explore the implicit relationships between items based on Linked Data and some measures for computing the semantic distances.

Ko et al. [6] proposed new approach that allows recommending potentially interesting content for users using semantic groups generated by each user viewing. The LOD were used in the generation of clusters by collecting more information about the films seen by users.

B. Content-based filtering

Pan et al. [7] uses Linked Data sets to recommend music artists based on the specified user interest. The system proved as effective when making discoveries of relevant artists.

Lasek [8] proposes a hybrid news articles recommendation system, which merges content processing techniques and data enrichment via LOD.

Di Noia et al. [9] and [3] are one of authors that have using content-based filtering. The system allows upgrading a recommendation system with linked data and using the Vector Space Model (VSM) and apply it to the recommendation of movies.

C. Hybrid filtering

Zarrinkalam and Kahani [10] propose in an approach that uses citation recommendation related to improving local data. This approach is a combination of filtering based on content and multi-criteria collaborative filtering.

Ostuni et al. [11] offer the best recommendations relative to implicit feedback by using Linked Data. This approach allows recommending items by exploiting their properties and attributes that are defined in a semantic graph.

The main idea of [12] is to automatically enricher the attributes of objects using Linked Data to improve the content-based recommendation.

Meymandpour and Davis [13] present a hybrid approach that combines the semantic analysis of items using LOD with collaborative filtering approaches. The proposed approach demonstrated a higher accuracy in comparison with a user-based collaborative filtering technique.

Alhamid et al. [14] introduce a hybrid collaborative context approach that use a physiological data to enhance the recommendation process and show the importance of using contextual information for the improvement of the quality of recommender system.

Abderrahim and Benslimane [15] propose a system for providing recommendation based on the Social Trust Network. The system combines Social Networks of users and Social Networks of Web service to provide recommendations for a target user. The provided recommendations are based not only on the similarity between users and between Web services, but also on the trust value of user and Web service.

III. The Proposed Approach

This section presents our approach in detail for semantics-enhanced recommender system. We describe the overall architecture of our system based on the LOD.

This architecture is divided into two parts. The first part, which is executed in offline mode, relates to the preparation of the data. It is divided in three steps: enrichment, clustering and generating of data model. The second part is executed on online mode concerning the recommendation. It include generating, filtering and ranking paths. The overall process is depicted in Fig. 1.

A. Resources

Three resources (U, I, E) that represent user, item and entities descript our recommender system. User is the person that uses the recommender system. He gives his opinion and he receives the recommendations. User is described by its profile (i.e. preferences).

Each user U_i is represented by (id, name, {(Keyword$_1$: value), (Keyword$_2$: value),.., (Keyword$_n$,: value)}).

Example 1

U_1= (365, Mohamed, {(Actress: Angelina Jolie), (Sports: football), (Movie: MI), (Singer: Celine Dion)}).

The item I is the term used to denote what the system should recommend to a user U. Each item I is represented by (id, name, {(Keyword$_1$: value), (Keyword$_2$: value),.., (Keyword$_n$: value)}).

Example 2

I_1= (128, Peugeot 206, {(Color: Black), (Energy: Diesel)}).

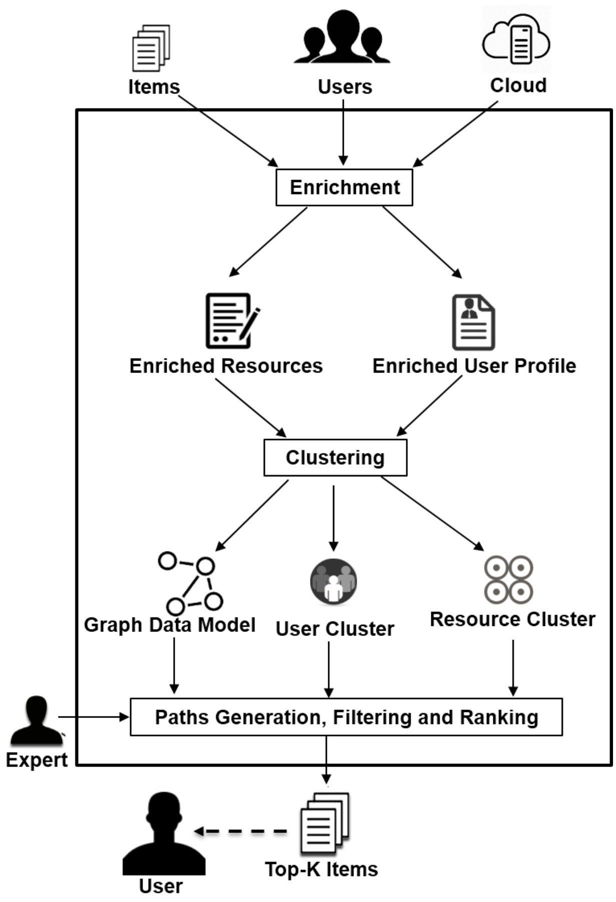

Fig.1. Semantics-enhanced recommender system

B. *Preparation of data*

The first part of our recommender system concerns the preparation of the data, it is divided in three steps: enrichment, clustering and generating of data model.

1. Enrichment

Enrichment increases the semantic description of data. Several semantic sources are used to ensure this enrichment (Metadata, ontologies, LOD, etc.).

DBpedia as a central, large and best-interlinked LOD data source fits good for this purpose. DBpedia is one of the main projects of the Linked Open Data community, which "focuses on the task of converting Wikipedia content into structured knowledge, such that Semantic Web techniques can be employed against it" [16]. Our system maintains one or more SPARQL Endpoints to various LOD datasets, and then it harvest the description of each item of the system from these endpoints.

In the same way, user's profiles have to be enriched. We obtain in output of this phase an enriched user's profile and an improved item's description.

Example 3

User U_1 of example 1, will be transformed after enrichment into U_1',

Where *U_1'= (365, Mohamed, {(Actress: Angelina Jolie), (Sports: Football), (Movie: MI), (Singer: Celine Dion), (Book: Harry Potter)}).*

In the same way, the item I_1 of example 2, will be transformed into I_1',

Where *I_1'= (128, Peugeot 206, {(Color: Black), (Energy: Diesel), (Power: 4CV)}).*

2. Clustering

In this phase, items will be grouped according to their semantic similarities. We calculate the similarity between semantic descriptions of items. Similarly, users will also be grouped using their profiles and evaluations.

Many clustering algorithms have been applied to document data sets. These algorithms include, but not limited to, K-means [17], Graph similarity [18], and VSM [19].

The purpose of K-means is to divide the observations into K clusters in which each observation belongs to the cluster with the nearest mean.

Graph similarity is a graph in which the nodes and edges represent its entities and the relationships between these entities.

VSM calculates a degree of similarity between the query and the document. The queries and documents are represented by the vectors of the weight of the words that form them. The degree of similarity is expressed as the similarity between the query vector and document vector.

In this paper, we have tested the three techniques.

3. Data Model Generation

The collected data from the enrichment phase will be used for the generation of data model. The data model is a semantic graph represented by two sub-graph that consist of nodes connected to each other through edges (Fig. 2).

The first sub-graph represents a contextual user feedback. Contextual plays an important role regarding the perception of the usefulness of an item for a user. It can greatly influence the recommendation accuracy.

Many researchers have worked on improving the quality of recommender systems by utilizing users' context.

Three context dimensions area exploited in this work: spatial, social and temporal (see Definition 1).

Definition 1. Contextual user feedback model

Formally, a contextual user feedback is modeled as G_1 (U, I, R_j^t) where U is a set of users, I a set of items, and R_j^t represents the feedback value (specified on a scale of 1 to 5) of user U on item I. *t* represents the context of the feedback (e.g., time, location), and *j* the value of the context (e.g, day or night for the context time).

The second sub-graph represents semantic item description (see Definition 2).

Definition 2. Semantic item description model

Formally, a semantic item description is modeled as G_2 (I, P, E), where I and E are respectively a set of items and semantic proprieties, and P represents the weight of the semantic properties.

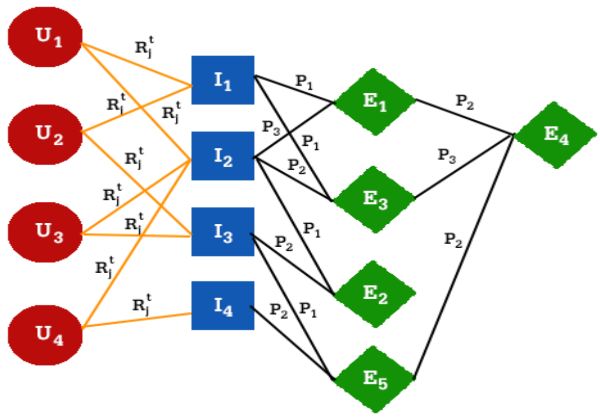

Fig.2. Representation of graph data model

C. *Recommendation*

The second part include generating, filtering and ranking paths. Top-K items targeted by the paths classified according to the path average are recommended to the user.

1. Generating and Filtering Paths

To recommend a new item I_j to a user U_i, we have to extract all the paths C_k that lead the user U_i to this item I_j.

Example 4

For the query (U_3, I_1), that checks if the item I_1 can be recommended the user U_3, nine paths will be extracted.

(U_3, i_3, e_2, i_2, U_1, I_1);
(U_3, i_2, U_1, I_1);
(U_3, i_3, U_2, I_1);
(U_3, i_2, e_3, e_1, I_1);
(U_3, i_3, e_2, e_4, e_1, I_1);
(U_3, i_3, e_5, e_4, e_1, I_1);
(U_3, i_2, U_4, i_4, i_3, U_2, I_1);
(U_3, i_3, i_4, e_5, e_4, e_1, I_1);
(U_3, i_3, e_2, i_2, e_3, e_1, i_2, U_1, I_1).

Once, all the paths extracted, we calculate the weight of each paths C_k and we retain those having an average greater than or equal to the threshold value given by the user or predefined by the recommender system.

The weight of a path is calculated using the weight of the properties connecting the vertices of a path.

Example 5

Assume that for Example 4, the weights of the properties are respectively P_1=3; P_2=1; P_3=2; P_4=5. After calculation, we obtain the following paths.

C1(5, 1, 3, 3, 2): 14/5= 2.8 ;
C2(1, 3, 2): 6/3=2 ;
C3(1, 1, 3): 5/3=1.6 ;
C4(1, 3, 2, 3): 10/4= 2,5 ;
C5(5, 1, 2, 1, 3): 12/5= 2,4 ;
C6(5, 3, 1, 1, 3): 13/5=2,6 ;
C7(1, 5, 4, 5, 2, 4): 21/6=3,5 ;
C8(5, 5, 1, 1, 1, 3): 16/6=2,67 ;
C9(5, 1, 3, 1, 2, 1, 3, 2): 18/8=2,25.

Considering the threshold= 2.5, we will retain respectively the paths C_4, C_6, C_7, and C_8.

2. Path Ranking

The results of the previous step are shown in Fig. 03.

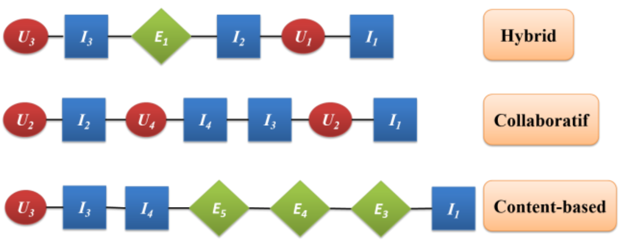

Fig.3. Filtering the resulting paths relative to the threshold

We can notice that there are three types of paths that will be treated differently: Content-based, Collaborative, and Hybrid.

Content-based paths

If the path is based on content (i.e., if there is in the path one user and items connected with entities), we check if the items that constitute the path are the in the same cluster.

Collaborative paths

If the path is collaborative (i.e., the path is a list of items and users), we check if the users that make up the path belong to the same cluster.

Hybrid paths

If the path is hybrid (i.e., in which there are items, users and entities), we check whether the two cases above are checked.

Finally, all items targeted by the paths satisfying the above conditions are classified according to the path average and recommended to the user.

IV. Experiments

We have conducted a couple of experiments in order to evaluate the quality of provided recommendations. In this section, we conduce the experimental setup and provide comprehensive analysis on the experimental results.

A. Experimental Setting

The experiments were performed with the following resources and conditions.

1. Data Set

We conducted several experiments on the popular Movie Lens dataset. The dataset, taken from the real world in the field of films, contains 1,000,209 ratings for 3,883 movies provided by 6,040 users. MovieLens datasets are mainly aimed at evaluating collaborative recommender systems in the movie domain. Since our approach is based on a content-based recommendation, in order to use such datasets to test the performances of our algorithms, we linked resources represented in MovieLens to DBpedia ones.

2. Computing Environment

The proposed approach was coded in java programming environment. The experiments were conducted on a desktop computer that runs under Windows 7 (64bit) with Intel Core i7-2600 CPU @3.40GHz, 8.00GB of RAM.

B. Results and Analysis

To demonstrate the effectiveness and the performance of LOD-based recommender system, we conducted several experiments. In the following, we present results of experiments conducted on the abode-synthesized dataset. In each experiment, we randomly choose twenty users as the target users for making recommendation.

1. Impact of User-graph Constructing Technique

To study the effect of the technique of constructing the graph of user provided by recommender system, we compare the results using alternately three techniques: 1) personal information, 2) ranking, and 3) hybridization of the two techniques. We notice that hybridization leads to better results (Fig. 4).

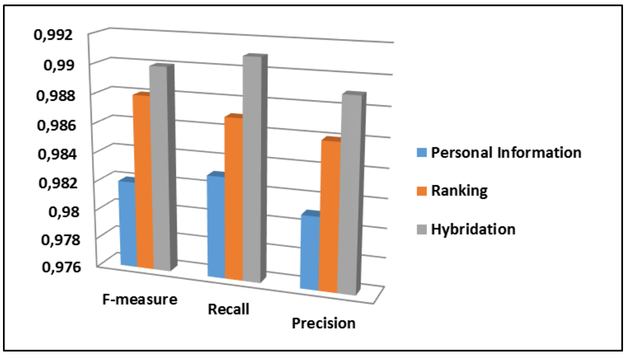

Fig.4. Evaluation of system with different methods to construct the graph of users.

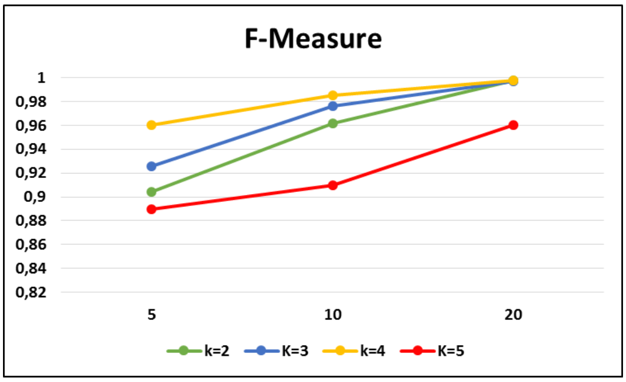

Fig.6. Impact of the number of cluster on recommendation

2. Parameter Impact on Clustering

To study the parameter impact on VSM clustering, we performed four tests while changing the values of three parameters of the Movie-Lens dataset, namely Genre, Actor and Writer. We take respectively the values:

VSM (genre=1, actor=1, writer=1)
VSM (genre=5, actor=3, writer=1)
VSM (genre=1, actor=5, writer=3)
VSM (genre=3, actor=1, writer=5)

The performance of VSM clustering is measured in terms of three external validity measures namely Recall, Precision and F-measure.

Fig. 5 shows that changing the settings of VSM leads to change the quality of recommendation. We notice that giving the same value to the three parameters generates the best results.

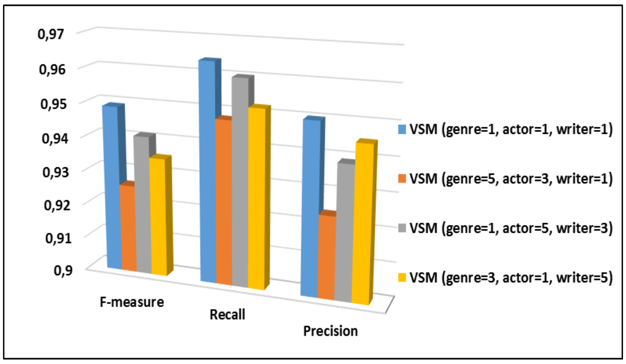

Fig.5. The evaluation of VSM.

3. Impact of the Number of Clusters K

Let us evaluate the impact of the number of clusters K on the characteristics of each cluster. We repeat the previous test with the number of clusters varying from 1 to 5.

In Fig. 6, we note that varying the number of clusters from 1 to 5, affects the quality of the recommendation. We deduce that the value 4 leads to better results. Based on this observation, we have chosen k=4 for the rest of the experiments.

4. Impact of the clustering technique

The main objective of this experiment is to explore the impact of the techniques used for clustering users and items on the quality of recommendation.

We perform four tests to compare between different techniques used for clustering users and items respectively.

First, we use K-means for clustering users and items. Second, we use K-means for clustering users and VSM for clustering items. Third, we use Graph similarity for clustering users and K-means for clustering items. Finally, we use Graph similarity for clustering users and VMS for clustering items. The performance of each clustering algorithm is measured in terms of the external validity measures F-measure.

As mentioned in Fig. 7, it is obvious that the best results are achieved while using K-means for clustering users and VSM for clustering items.

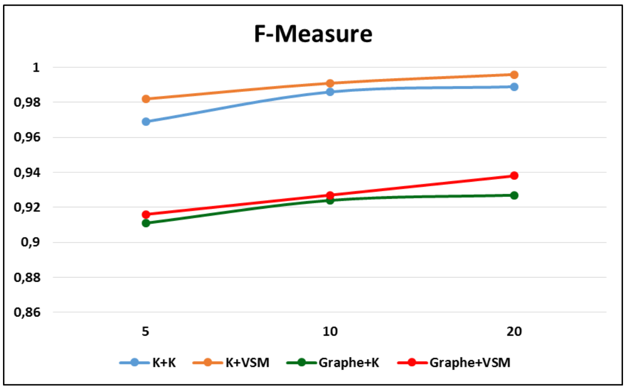

Fig.7. Comparison between different clustering techniques.

5. Impact of Using LOD on Improving Recommendations

In this phase of experiment, our aim is to demonstrate that adding semantic information from LOD can improve recommender systems capability. For each pair of objects visited by the same user, we have computed content based similarity either including or not including DBPedia data.

In Fig. 8, we note that the use of LOD helps in the improvement of the recommendation and gives the best results.

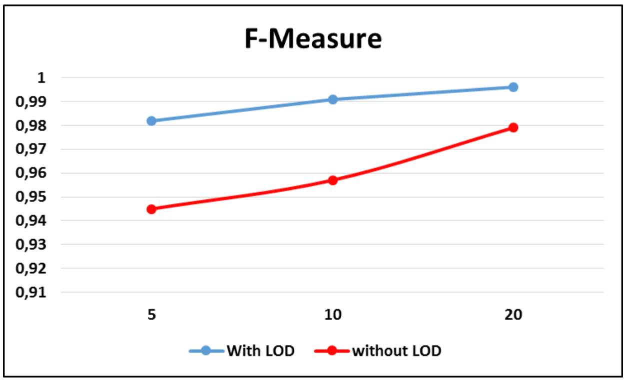

Fig.8. The influence of LOD on recommendation quality.

6. Impact of Using Context

In context-aware content-based recommendations, the content-based algorithm is extended to take into account the context of the user. Before generating the recommendations, the user feedback is processed by a contextual pre-filter. Contextual information is used to determine the relevance of the feedback and filter these data based on the current situation.

To compare the contextual and un-contextual approaches, we conducted experiments across the following experimental settings: 1) Contextual user feedback model Context Model, 2) Un-contextual user feedback model.

In Fig. 9, we note that contextual information affects positively the quality of the recommendations.

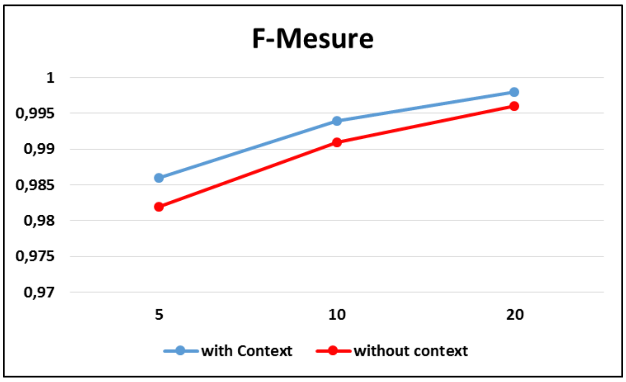

Fig.9. The influence of context on the provided recommendations.

7. Global Comparison

The goal of this experiment is to explore the quality of generated recommendations by their comparison with existing recommendation systems. The measurements performed in this experiment were done by taking into consideration, all the findings mentioned in the previous tests.

While comparing our approach with [9] and [11], we notice that our approach surpasses its competitors.

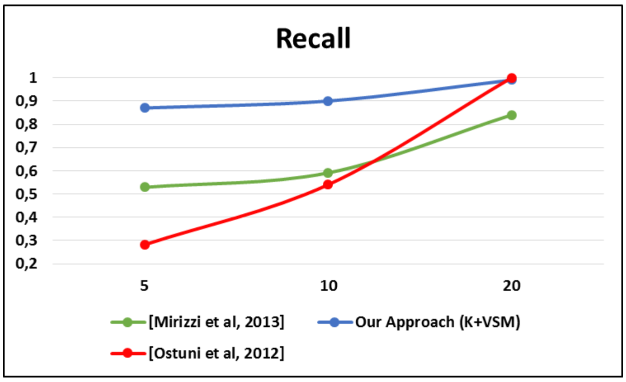

Fig.10.Comparison of approaches

V. Conclusion

This paper demonstrated the applicability of LOD for improving the accuracy and the quality of recommendation systems. The experiments showed that the accurate measurement of item similarities using LOD has the potential to improve the performance of recommender systems, especially, in situations where an insufficient amount of user ratings is available. The combination of semantic enrichment of items and users with collaborative filtering-based recommendation in the proposed hybrid recommender system presented comparable overall accuracy, in addition to significant improvement in resolving the item cold-start and data sparsity problems.

For future work, we plan to improve the approach in several aspects. This includes using others resources of linked data (e.g., DBTrope, Freebase and LinkedMDB) and others datasets (e.g., LastFM). We also plan make hybridization between implicit and explicit feedback.

Acknowledgment

The work described in this paper was supported by the National Research Project of Algeria under grant No. C00L07UN220120130004.

References

[1] N. Abderrahim, S.M. Benslimane, 2015. Towards Improving Recommender System: A Social Trust-aware Approach. International Journal of Modern Education and Computer Science (IJMECS), 7(2):8-15.

[2] R. Burke, 2007. Hybrid web recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, The Adaptive Web: Methods and Strategies of Web Personalization, volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York.

[3] Tommaso Di Noia, Vito Claudio Ostuni, 2015. Recommender Systems and Linked Open Data. Reasoning Web. Web Logic Rules. LNCS 9203, pp 88-113.

[4] Benjamin Heitmann and Conor Hayes, 2010. Using Linked Data to Build Open, Collaborative Recommender Systems, In AAAI Spring Symposium: Linked Data Meets Artificial Intelligence.

[5] Rui Yang, Wei Hu, and Yuzhong Qu, 2012. Using Semantic Technology to Improve Recommender Systems Based on Slope One, CSWS.
[6] Han-Gyu Ko, Eunae Kim and In-Young Ko, 2014. Semantically based Recommendation by using Semantic Clusters of Users' Viewing History.
[7] Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B, 2010. Music Recommendations Using DBpedia. In ISWC'10.
[8] Lasek, I., 2011. Dc proposal: Model for news filtering with named entities. In The Semantic Web – ISWC 2011, LNCS, vol. 7032, pp. 309–316.
[9] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito and Markus Zanker, 2012. Linked Open Data to support Content-based Recommender Systems. In the 8th International Conference on Semantic Systems, Graz, Austria.
[10] Fattane Zarrinkalam, Mohsen Kahani, 2012. A Multi-Criteria Hybrid Citation Recommendation System Based on Linked Data, In Computer and Knowledge Engineering (ICCKE).
[11] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, Roberto Mirizzi, 2013. Top-N Recommendations from Implicit Feedback leveraging Linked Open Data, In the 7th ACM conference on Recommender Systems, RecSsys.
[12] Ladislav Peska and Peter Vojtas, 2013. Enhancing Recommender System with Linked Open Data.In the International Conference on Flexible Query Answering Systems, Granada, Spain.
[13] Rouzbeh Meymandpour and Joseph G. Davis, 2015. Enhancing Recommender Systems Using Linked Open Databased Semantic Analysis of Items. In the Proceedings of the 3rd Australasian Web Conference (AWC 2015), Sydney, Australia.
[14] M. F. Alhamid, M. Rawashdeh, H. Al Osman, M. Shamim Hossain, A. El Saddik, 2014. Towards context-sensitive collaborative media recommender system, International Journal of Multimedia Tools and Applications, 74(24):11399–11428.
[15] Auer, S., C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives (2007). DBpedia: A Nucleus for a Web of Open Data. The Semantic Web, Lecture Notes in Computer Science 4825, 722–735.
[16] Naziha Abderrahim, Sidi Mohamed Benslimane. Towards Improving Recommender System: A Social Trust-aware Approach. International Journal of Modern Education and Computer Science (IJMECS), Vol.7, No. 2, January 2015.
[17] MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, vol. 1, pp. 281–297. University of California Press, California (1967)
[18] Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620. 1975.

Authors Profiles

Asmaa Fridi is a PhD student in computer science Department at the University of Sidi Bel-Abbes, Algeria. She received a M.S. degree in computer science in 2011 from the Computer Science Department of Sidi Bel Abbes University. Her research interests include web services, social networks, semantic web, and recommender systems.

Sidi Mohamed Benslimane is a full Professor at École Supérieure en Informatique, Sidi Bel-Abbès, Algeria. He received his PhD degree in computer science from Sidi Bel Abbes University in 2007. He also received a M.S. and a technical engineer degree in computer science in 2001 and 1994 respectively from the Computer Science Department of Sidi Bel Abbes University, Algeria. He is currently Head of Higher School of Computer Science, Sidi Bel-Abbès, Algeria. From 2001 to 2015, he was a member of the Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS. Actually, he heads the Research Team 'Service Oriented Computing' at LabRI-SBA Laboratory. His research interests include, semantic web, service oriented computing, ontology engineering, information and knowledge management, distributed and heterogeneous information systems and context-aware computing.

How to cite this paper: Asmaa Fridi, Sidi Mohamed Benslimane,"Towards Semantics-Aware Recommender System: A LOD-Based Approach", International Journal of Modern Education and Computer Science(IJMECS), Vol.9, No.2, pp.55-61, 2017.DOI: 10.5815/ijmecs.2017.02.07

