International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

A New Semantic Similarity Measure Based on Ontology for Movie Rate Prediction

Vandana Mohan Patil, J. B. Patil

Abstract: ***A recommendation algorithm comprises of two important steps: 1) Predicting rates, and 2) Recommendation. Rate prediction is a cumulative function of the similarity score between two movies and rate history of those movies by other users. There are various methods for rate prediction such as weighted sum method, regression, deviation based etc. All these methods rely on finding similar items to the items previously viewed/rated by target user, with assumption that user tends to have similar rating for similar items. Computing the similarities can be done using various similarity measures such as Euclidian Distance, Cosine Similarity, Adjusted Cosine Similarity, Pearson Correlation, Jaccard Similarity etc. All of these well-known approaches calculate similarity score between two movies using simple rating based data. Hence, such similarity measures could not accurately model rating behavior of user. In this paper, we will show that the accuracy in rate prediction can be enhanced by incorporating ontological domain knowledge in similarity computation. This paper introduces a new ontological semantic similarity measure between two movies. For experimental evaluation, the performance of proposed approach is compared with two existing approaches: 1) Adjusted Cosine Similarity (ACS), and 2) Weighted Slope One (WSO) algorithm, in terms of two performance measures: 1) Execution time and 2) Mean Absolute Error (MAE). The open-source Movielens (ml-1m) dataset is used for experimental evaluation. As our results show, the ontological semantic similarity measure enhances the performance of rate prediction as compared to the existing-well known approaches.***

Keywords: ***Ontology, Recommendation, Rate prediction, Semantic similarity measure.***

I. INTRODUCTION

Rate prediction is an important milestone in recommendation process [1]. The problem of rate prediction can be stated as: The rate prediction of item i by user u is dependent on ratings given by user u to other similar items to i. Thus, similarity calculation plays a vital role in rate prediction [1]. Numbers of techniques exist for computing similarity measures such as Euclidian Distance, Cosine Similarity, Adjusted Cosine Similarity, Pearson Correlation, Jaccard Similarity etc. All of these approaches calculate similarity score between two movies using simple rating based data**.** Hence, such similarity measures could not accurately model rating behavior of user. This paper shows that the accuracy in rate prediction can be enhanced by using ontology to calculate the similarity between two movies at conceptual level. *Ontology is an explicit specification of a conceptualization* [2]. Ontology provides a set of well-founded constructs that define significant concepts and their semantic relationships. Domain ontologies usually include concepts, subsumption relations between concepts (concept hierarchies), and other relationships among concepts that exist in the domain [3]. Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains. Number of approaches has been proposed to integrate domain knowledge in recommender systems in the form of ontology [4]. Ontologies are commonly handled as concepts hierarchies with attributes and relations, which set up a terminology to define semantic networks of interrelated concepts and instances. In general, when a domain model is represented as ontology, items and user models consist of a subset of concepts from the domain ontology [4]. Ontologies can be constructed automatically, semi-automatically and manually.

II. RELATED WORK

Maedche *et. al.* [5] introduced three types of similarity measures: 1) Taxonomy similarity (TS), 2) Relation similarity (RS), 3) Attribute similarity (AS). Taxonomy similarity between two instances (TS) is based on their corresponding concepts' positions in concept taxonomy which is defined in ontology model. Suppose that comedy and horror are sub-concepts of movie concept. Movie A and Movie B are comedy movies and Movie C is a horror movie. In such an ontology-based metadata, taxonomy similarity between Movie A and Movie B is higher than the taxonomy similarity between Movie A and Movie C. The second type of similarity measure using ontology based metadata is relation similarity. Relation similarity (RS) between two instances is based on their relations to other instances in ontology. Suppose that Director X is the director of Movie A and Movie B, Director Y is the director of Movie C. In this example, relation similarity between Movie A and Movie B is higher than the one between Movie A and Movie C because director of Movie A and Movie B is the same. Attribute similarity (AS) is the third similarity measure. The attribute similarity between two instances depends on their attribute values. Mobasher *et.al.* [6] proposed an approach for semantically

Manuscript published on 30 September 2019
* Correspondence Author
Vandana Mohan Patil*, Department of Computer Engineering, SES's R.C.Patel Institute of Technology, Shirpur Dist. Dhule, India. Email: shraddhasam5581@gmail.com
Prof. Dr. J. B. Patil, Department of Computer Engineering, SES's R.C.Patel Institute of Technology, Shirpur Dist. Dhule, India. Email: jbpatil@hotmail.com

enhanced collaborative filtering. Structured semantic knowledge about items, extracted automatically from the Web based on domain-specific reference ontologies, is used in conjunction with user-item mappings to create a combined similarity measure and generate predictions. Schickel *et.al.* [7] defined a novel similarity measure called Ontology Structure based Similarity (OSS). It is based on assigning a-priori score (APS) to concepts in the ontology, and computing the relations between the scores assigned to different concepts. These similarities are then used to propagate scores for a specific user. Ceylan *et.al.* [8] proposed a hybrid approach called SEMCBCF, that uses semantic similarities between items using ontology-based metadata and content-based user models in the movie domain. By using ontology-based meta data and user models, the system finds the similarities between items and recommends items to users using similarities between items and users explicit ratings. In SEMCBF, similarities between items are calculated using three types of similarity measures which are described by Maedche *et. al.* [5] i.e. Taxonomy similarity (TS), Relation similarity (RS) and Attribute similarity (AS). Groues *et.al.* [9] presented an extension of a semantic similarity measure-a version of the Maedche and Zacharias similarity measure, for semantic instances to be used with real linked data sources such as DBpedia [9].

III. CONVENTIONAL RATE PREDICTION

Rate prediction is a cumulative function of the similarity score between two movies and rate history of those movies by other users. There are various methods for rate prediction such as weighted sum method, regression, deviation based etc. All these methods rely on finding similar items to the items previously viewed/rated by target user; with assumption that user tends to have similar rating for similar items. Computing the similarities can be done using various similarity measures such as Euclidian Distance, Cosine Similarity, Adjusted Cosine Similarity, Pearson Correlation, Jaccard Similarity etc. For experimental evaluation, we have implemented two conventional approaches *viz* Adjusted Cosine Similarity, and Weighted Slope One algorithm. The Adjusted Cosine Similarity is a similarity based approach used along with weighted sum method of rate prediction, while Weighted Slope One is deviation based approach. All these techniques predict the rate of an item by active user using the ratings given to the similar items by other users, who have rated both the movies [10].

A. Adjusted Cosine Similarity (ACS)

Adjusted cosine similarity (ACS) [8] is a similarity measure based on historical rating data only. It is used along with the weighted sum rate prediction method. It assumes that different users have different rating tendency. The adjusted cosine similarity measure of item *i* and item *j* is a function of rating to item *i* and *j* by all the users who rated both the items and mean rating of the user. It is given by,

$$Sim_{i,j} = \frac{\sum_{u \in U} (R_{u,i} - \overline{R_u})(R_{u,j} - \overline{R_u})}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R_u})} * \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_u})}} \quad (1)$$

where-

- U is set of users who rated both items *i* and *j*.
- $Sim_{i,j}$ is similarity score between items *i* and *j*.
- $R_{u,i}$ is Rate of user *u* to item *i*.
- R_u is Mean Rating of item *u*.

Weighted Sum Method for Rate Prediction

The weighted sum method [10] relies on the similarity computation between two items. Similarity measure between item *i* and *j* can be computed using different techniques *for example* Pearson correlation, cosine similarity, adjusted cosine similarity, etc. [10]. The weighted sum method for rate prediction is given as:

$$Pr_{u,i} = \frac{\sum_{j \in similarTo(i)} S_{i,j} * R_{u,j}}{\sum_{j \in similarTo(i)} S_{i,j}} \quad (2)$$

where,

- $Pr_{u,i}$ - Predicted rate of item *i* by user *u*.
- *similarTo(i)* - Set of items similar to item *i*.
- $S_{i,j}$ - Similarity measure between item *i* and *j* where $j \in similarTo(i)$.
- $R_{u,j}$ - Rate given by user *u* to item *j* where $j \in similarTo(i)$.

B. Weighted Slope One Algorithm (WSO)

The weighted slope one algorithm [11] is variation of slope one algorithm [11]. The algorithm works in two steps:

1. Compute deviation between every pair of items i.e. $dev_{i,j}$. It is calculated as-

$$dev_{i,j} = \sum_{u \in S_{i,j}(X)} \frac{u_i - u_j}{card(S_{i,j}(X))} \quad (3)$$

where-

- $dev_{i,j}$ is deviation between items *i* and *j*.
- X is the entire set of all ratings.
- $card(S_{i,j}(X))$ is number of people who rated both the items *i* and *j*.

2. Make rate predictions using deviations computed in Step1. It is calculated as-

$$Pr_{u,i}^{ws1} = \frac{\sum_{j \in S(u)-\{i\}} (dev_{i,j} + u_j) * card(S_{i,j}(X))}{\sum_{j \in S(u)-\{i\}} card(S_{i,j}(X))} \quad (4)$$

where-

- $Pr_{u,i}^{ws1}$ is prediction of user *u*'s rating on item *i*.
- $card(S_{i,j}(X))$ i.e. number of people who rated both the items *i* and *j*.
- $dev_{i,j}$ is deviation between items *i* and *j*.
- $j \in S_{(u)-\{i\}}$ is every item *j* that user *u* has rated except *i*.
- u_j is rate of user *u* to item *j*.

IV. ONTOLOGY BASED RATE PREDICTION

Rate predictions based on conventional methods are less accurate due to lack of domain knowledge. Incorporating domain knowledge into rate prediction to improve the accuracy is one of the potential research areas from last few years. Ontologies are the dominant way for representing domain knowledge in lots of areas, and hence, a number of approaches have been proposed to integrate ontological knowledge in recommender systems [4].

Ontologies enable semantic interpretation of terms [12]. In this section, the proposed approach is explained in detail.
The proposed approach uses the weighted sum method of rate prediction. This method relies on similarity measure between two items. Here, we have introduced a new semantic similarity measure named as, *"Ontology Based Semantic Similarity (OBSS)"*.

Framework of Proposed Approach

Our proposed system consists of four phases:
1. Ontology Construction
2. Data Preprocessing
3. Pattern Discovery
4. Pattern Analysis

A. Ontology Construction

First step is construction of domain ontology. The process of acquiring, maintaining and enriching the domain ontologies is referred to as *"Ontology Engineering"* [3].Ontologies can be constructed manually, semi-automatically and automatically [3]. We have manually constructed domain ontology i.e. *Movie ontology* using the notion of aggregate representations for groups of objects that have homogeneous concept structure. Such group of objects is called as a *'class'*. A class *C* is a set of objects along with set of *'attributes'*. These attributes define the internal properties of the objects in C as well as the relationships with other objects in *C* [3]. Fig. 1 shows the Movie ontology. As shown in Fig. 1, the Movie ontology has one top level concept 'Movie'. It has attributes *viz MovieID, Title, Genres, Year, Tags and Rating.* All these attributes are related to concept *'Movie'* with *'has_Attribute'* relationship. In Loh *et.al.* [13], a concept is defined as *"a group of terms that are semantically relevant"*. Using this concept definition, we will find interesting concept/conceptual patterns. As shown in Fig.1, there are 19 distinct genres *viz 'Action', 'Adventure', 'War', 'Comedy', 'Family', 'Children', 'Documentary', 'Sci-Fi', 'Animation', 'Drama', 'I-MAX', 'Crime','Fantasy', 'Thriller', 'Mystery', 'Horror', 'Film-Noir', 'Romance', 'Musical', 'Western'*. For the proposed approach, these genres are classified into six disjoint types with the notion of interest-type of a user. e.g. 'Action', 'War' and 'Crime' show similarity in interest types i.e. if a user likes a movie with

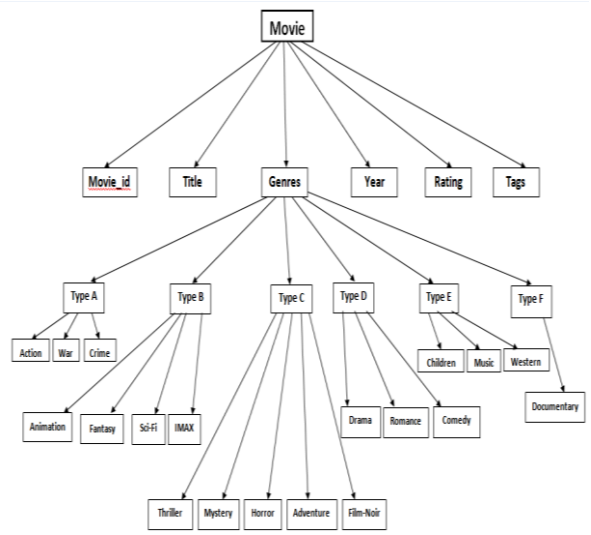

Fig. 1. Movie Ontology

Fig. 2.

'Action' genre, then he/she will also like movies with 'Crime' or 'War' genres *i.e.* these concepts are semantically relevant. Without domain semantics, 'Action' and 'War' genres are treated as completely unrelated terms; whereas using ontological domain knowledge, these two genres are treated as similar at conceptual level, as they belong to same concept. All genres are related to their types with *'belongs_To'* relationship and all genre-types are related to *'Genres'* with *'is-a'* relationship.

B. Data Preprocessing

The data in dataset may not be in the proper format that is to be directly used by further phases. Hence, data preprocessing phase transforms the raw data in the format that can be processed by further phases. The data preprocessing consists of three steps: 1) Data cleaning, 2) Map the genres, and 3) Item profile construction.

- **Data cleaning**

Data cleaning is one of the common data preprocessing steps that remove the unnecessary data [14]. The *'timestamp'* column is dropped from ratings database, as it is not required for our system. Further, the movie database contains three fields *viz MovieID, Title, Genres.* Table I shows the sample of the movie database. In preprocessing step, we have removed the '|' symbol and separated the genres in *Genre* field into individual genres as shown in Table II.

Table-I: Sample Movie Database

Movie ID	Title	Genres
1	Toy Story (1995)	Animation | Children | Comedy
89	Nick of Time (1995)	Action | Thriller
116	Anne Frank Remembered (1995)	Documentary
149	Amateur (1994)	Crime | Drama | Thriller
258	Kid in King Arthur's Court (1995)	Adventure | Children | Comedy | Fantasy | Romance

Table-II: Pre-processed Sample Movie Database

Movie ID	Title	genre1	genre2	genre3	genre4	genre5
1	Toy Story (1995)	Animation	Children	Comedy		
89	Nick of Time (1995)	Action	Thriller			
116	Anne Frank Remembered (1995)	Documentary				
149	Amateur (1994)	Crime	Drama	Thriller		
258	Kid in King Arthur's Court (1995)	Adventure	Children	Comedy	Fantasy	Romance

- **Mapping of Genres**

In this phase, we will transform the physical patterns into conceptual patterns of user rating behavior. For this, the genre-list for each movie in our dataset is mapped to their top level concepts as described by our *Movie* ontology i.e. from physical level to conceptual level which is more

generalized. Table III shows all the genre-types of our *Movie* ontology. Algorithm 1 describes the mapping process and Table IV shows the mapped genres of Table II.

Let,

M is set of all movies, M= $\{m_1, m_2, ..., m_p\}$,

G is set of genres for a movie in M, G= $\{g_1, g_2, ..., g_q\}$,

C is set of genre-types, C={'A','B','C','D','E','F'}.

Algorithm 1: Mapping of genres

Input: M,G,C

Output: List of movies with mapped-genres

Processing:

for each movie *i*in M

for each genre *j* in G

Map *j* to its top level concept in C

These mapped genres are then fed as input to the *Item Profile Construction* step to construct bitmap of each movie.

- **Item Profile Construction**

In the proposed approach, the *Item profile* of each movie is a binary vector of 6 bits corresponding to 6 genres-types. Item profile for each movie is generated using the *binary scoring method* [15]. It is one of the scoring methods used in *Bag of Words (BOW)* model [15]. The binary scoring method simply marks *absence* and *presence* of word as *0* and *1* respectively. For each movie, the respective bit will be set to '1' if that movie's mapped-genres contain the type. For example, consider the movie 'Amateur (1994)' with mapped-genres A, D, C. So the binary vector for this movie is [1 0 1 1 0 0]. Algorithm 2 describes the item profile construction process and Table V gives the item profiles in Table IV. These item profiles are further used to compute the semantic similarity between two movies.

Let,

M is set of all movies, M= $\{m_1, m_2, ..., m_p\}$,

G is set of mapped-genres for a movie in M, G = $\{g_1, g_2, ..., g_q\}$,

C is set of genre-types, C= {'A','B','C','D','E','F'}

MBMP is set of movies with binary vectors. Each element in MBMP consists of Movie-ID, Title, A, B, C, D, E, F.

Table-III: Classification of Movie Genres

Sr.No.	Types	Genres
1	Type A	Action, War, Crime
2	Type B	Animation, Fantasy, Sci-Fi, IMAX
3	Type C	Thriller, Mystery, Horror, Adventure, Film-Noir
4	Type D	Drama, Romance, Comedy
5	Type E	Children, Musical, Western
6	Type F	Documentary

Table-IV: Sample Movie Database with mapped-genres

Movie ID	Title	genre1	genre2	genre3	genre4	genre5
1	Toy Story (1995)	B	E	D		
89	Nick of Time (1995)	A	C			
116	Anne Frank Remembered (1995)	F				
149	Amateur (1994)	A	D	C		
258	Kid in King Arthur's Court (1995)	C	E	D	B	D

Algorithm 2: Item Profile Construction

Input: M,G,C

Output:

MBMP

Processing:

for each movie *i* in M

for each genre *j* in G

for each concept type *c* in C

if *j* = *c*

then MBMP(*c*)=1

else MBMP(*c*)=0

Table V: Item Profiles

MovieID	Title	A	B	C	D	E	F
1	Toy Story (1995)	0	1	0	1	1	0
89	Nick of Time (1995)	1	0	1	0	0	0
116	Anne Frank Remembered (1995)	0	0	0	0	0	1
149	Amateur (1994)	1	0	1	1	0	0
258	Kid in King Arthur's Court (1995)	0	1	1	1	1	0

C. Pattern Discovery

The pattern discovery phase generates the usage patterns using historical rating data. Domain ontologies can be incorporated into pattern discovery phase to generate semantic usage patterns [3]. The proposed approach incorporates the ontological domain knowledge in this phase to generate semantic usage patterns. This process is discussed here in detail. The pattern discovery phase of the proposed approach involves: 1) Computing the Ontology Based Semantic Similarity (OBSS) measure between the movies, and 2) Predicting the rates of movies for the target user.

- **Ontology Based Semantic Similarity (OBSS) Calculation**

The semantic similarity between two movies is computed using the item profiles *i.e.* binary vectors of the movies. The binary vectors of the movies under consideration are compared bitwise to calculate the semantic similarity.

If all bits match, the semantic similarity is 1, if 2 out of 6 bits matched, the semantic similarity is 2/6 i.e. 0.33 and if no bit matches, the semantic similarity is 0. Thus, our proposed semantic similarity measure ranges from 0 to 1.

Algorithm 3 describes the ontology based semantic similarity calculation between two movies m_i and m_j. Let,

B_i and B_j represent the binary vectors of the movies m_i and m_j respectively.
N represents the number of bits similar between the binary vectors of the movies.

Algorithm 3: Calculating the ontology based semantic similarity between two movies m_i and m_j.

Input: B_i, B_j, N = 0
Output: *OBSS* measure between m_i and m_j *i.e.* $OBSS_{i,j}$

Processing:
for each bit b in B_i and B_j
if $m_i[b] = m_j[b]$
then N =

Table-VI shows an example of the ontology based semantic similarity score of the movie 'GoldenEye (1995)' with the movies in Table-V.

V. EXPERIMENTAL EVALUATION

The proposed approach is implemented in Python, and its accuracy is evaluated on the open source *MovieLens (ml-1m)* dataset *(http://www.grouplens.org)*. According to Cremonisi *et.al.* [16], to test the prediction accuracy, the original dataset should be properly partitioned into training and testing datasets. The training dataset is used to learn the model and the testing dataset is used to assess the quality of the model. These sets might have diverse proportions. This is the *Holdout method of data partitioning* [16]. The experiments are conducted on the browsing history of the current user i.e. *active session* and had split it into 80% training and 20% test data. The rates are predicted for every instance of test dataset using conventional approaches as well as using our proposed approach. The experiments are conducted with different users for variable number of movies in training and testing datasets.

Table-VI: Ontology Based Semantic Similarity (OBSS)Score

MovieID	Title	Semantic Similarity Score
1	Toy Story (1995)	0.17
89	Nick of Time (1995)	1
116	Anne Frank Remembered (1995)	0.5
149	Amateur (1994)	0.83
258	Kid in King Arthur's Court (1995)	0.33

Once the similarities are calculated, the rates for movies by active user are predicted using the weighted sum method [10].

Performance Evaluation Metrics

Two metrics are used to evaluate the performance of rate prediction, *viz execution time* required for similarity measure computation and *Mean Absolute Error (MAE)*.

- Execution Time

The experiments are executed on Core I3@2GHz computer system with 4GB RAM and 1TB HDD. The execution time required for similarity measure computation is a desirable feature for real time systems and it also increases the scalability of the system. Table VII and Fig. 2 shows the comparative performance of the approaches in terms of execution time required for similarity measure calculation.
From Fig. 2, it is clear that execution time required by the proposed approach is very less as compared to other approaches. This is due to less mathematical calculations involved in the proposed approach.

- Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a statistical accuracy metric [17]. It is the most popular and commonly used metric to evaluate the accuracy of rate prediction. It is calculated through direct comparison of the predicted ratings with the actual rating given by the user. It is a measure of deviation between predicted rating p_i and actual rating r_i [17]. It is calculated as follows [18]:

$$MAE = \frac{1}{N}\left(\sum_{i=1}^{N} | p_i - r_i |\right) \quad (5)$$

Table VIII and Fig. 3 shows the relative performance of the approaches in terms of MAE; lesser the value of MAE, more the accuracy in prediction. From Fig. 3, it is clear that the proposed approach gives more accurate results in rate prediction as compared to other approaches. Thus, proposed approach improves the accuracy of rate prediction based on the newly introduced semantic similarity measure.

Table-VII: Execution Time (*in seconds*)

Train size	WSO	ACS	OBSS
20	3.57	2.97	0.29
40	9.99	9.88	1.24
60	26.84	25.01	2.71
80	24.08	25.39	3.24
100	49.74	40.5	5.41
120	72.48	59.22	8.7
140	143.4	149.71	21.47
160	182.92	194.36	16.86
180	225.009	215.68	43.94
200	354.23	300.82	76.45
220	236.21	297.97	54.86
240	318.4	284.1	72.55

Table-VIII MAE (*Mean Absolute Error*)

Test size	WSO	ACS	OBSS
5	1.07	1.3	0.88
10	0.6	0.76	0.6
15	0.547	0.7	0.7
20	0.416	0.61	0.59
25	0.63	1.07	0.8
30	0.509	0.6	0.5
35	0.9	1.0	0.9
40	0.44	0.73	0.59
45	0.95	0.95	0.9
50	0.9	0.9	0.9
55	0.62	0.709	0.65
60	0.628	0.8	0.78

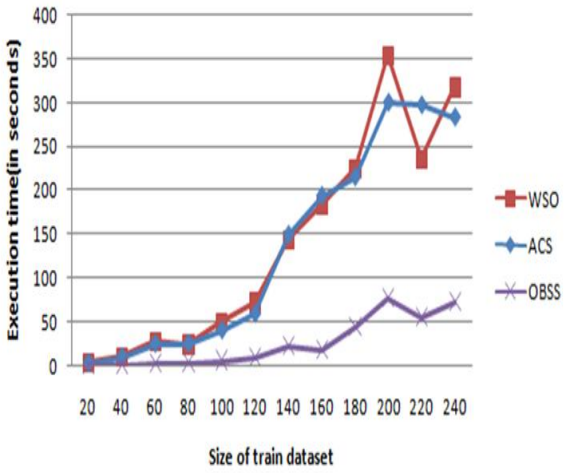

Fig. 2. Execution Time

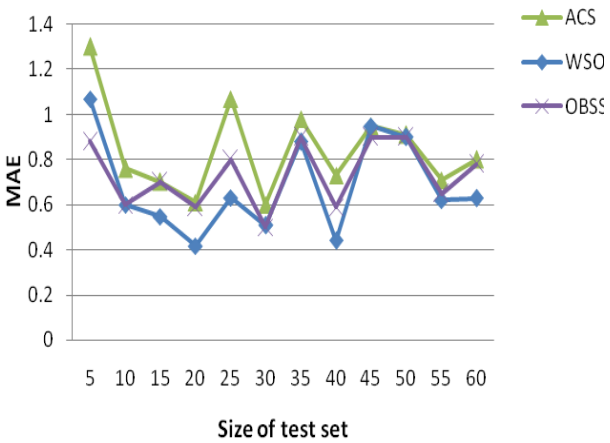

Fig. 3. Mean Absolute Error (MAE)

VI. CONCLUSIONS AND FUTURE WORK

All of the existing-well-known approaches calculate similarity score using simple rating based data. Hence, such similarity measures could not accurately model rating behavior of user. In this paper, we had introduced a new similarity measure named as *"Ontology Based Semantic Similarity (OBSS)"*. We had implemented a rate prediction approach using this semantic similarity measure for movies. Our approach computes the similarity between two movies using the ontological domain knowledge and emphasizes on the extent of similarity between two movies at conceptual level. From the experimental results, we can say that the performance of rate prediction can be enhanced through integration of domain knowledge in the form of ontology. As the proposed approach involves very less mathematical calculations, the execution time required for computation of ontology based similarity measure is very less even with huge number of similar items. So, the approach can deal with scalability problems. As well as, the accuracy of rate prediction is also improved with the semantic similarity measure. In future work, we will focus on pattern analysis phase, where these rate prediction data will be used for recommending *top-N* movies to the active user.

REFERENCES

1. Huseyin POLAT, Edip SENYUREK, "Effects of similarity measures on the quality of predictions", 26(4), Gazi University Journal of Science, 2013, pp. 557–562.
2. Thomas R. Gruber, "A translation approach to portable ontology specifications", 5(2), Knowledge Acquisition, 1993, pp. 199–220.
3. Honghua Dai, "Integrating semantic knowledge with web usage mining for personalization", In Web mining: Applications and techniques, IGI Global, 2005, pp. 276–306.
4. Marco De Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro, "Semantics- aware content-based recommender systems", In Recommender Systems Handbook, Springer, 2015, pp. 119–159.
5. Alexander Maedche and Valentin Zacharias, "Clustering ontology-based metadata in the semantic web", vol. 2431, 2002, pp. 348–360.
6. Bamshad Mobasher, Xin Jin, and Yanzan Zhou, "Semantically enhanced collaborative filtering on the web". In European Web Mining Forum, Springer, 2003, pp. 57–76.
7. Vincent Schickel-Zuber and Boi Faltings, "Inferring user's preferences using ontologies", Technical report, 2006.
8. Ugur Ceylan and Aysenur Birturk, "Combining feature weighting and semantic similarity measure for a hybrid movie recommender system", In The 5th SNA-KDD Workshop11, 2011.
9. Valentin Groues, Yannick Naudet, and Odej Kao, "Adaptation and evaluation of a semantic similarity measure for dbpedia: A first experiment", In 2012 Seventh International Workshop on Semantic and Social Media Adaptation and Personalization, IEEE, 2012, pp. 87–91.
10. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms", In Proceedings of the 10th International Conference on World Wide Web, WWW '01, New York, NY, USA , ACM, 2001, pp. 285–295.
11. Daniel Lemire and Anna Maclachlan, "Slope one predictors for online rating-based collaborative filtering", In Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 471–475.
12. David S´anchez, Montserrat Batet, David Isern, and Aida Valls, "Ontology-based semantic similarity: A new feature-based approach", Expert Syst. Appl., 39(9), July 2012, pp. 7718–7728.
13. Stanley Loh, Leandro Krug Wives, and Jos´e Palazzo M de Oliveira, "Concept-based knowledge discovery in texts extracted from the web", ACM SIGKDD Explorations Newsletter, 2(1), 2000, pp. 29–39.
14. Jason Brownlee, "How to Prepare Data For Machine Learning", [Online].Available: https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/ , 2013
15. Jason Brownlee, "A Gentle Introduction to the Bag-of-Words Model", [Online].Available: https://machinelearningmastery.com/gentle-introduction-bag-words-model/ , 2017.

16. Paolo Cremonesi, Roberto Turrin, Eugenio Lentini, and Matteo Matteucci, "An evaluation methodology for collaborative recommender systems", In 2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution, IEEE, 2008, pp. 224–231.
17. Mingang Chen and Pan Liu, "Performance evaluation of recommender systems", International Journal of Performabilty Engineering, 13(8), 2017, pp. 1246–1256.
18. F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, 16(3), 2015, pp. 261 – 273.

AUTHORS PROFILE

Ms. Vandana Mohan Patil is a Research Scholar, pursuing her Ph.D. in Computer Engineering. She is working as a Faculty in Department of Computer Engineering at R. C. Patel Institute of Technology, Shirpur. She is a recognized Post Graduate Teacher in the subject of Computer Engineering of North Maharashtra University, Jalgaon. Her research areas are Ontology, Data mining, Semantic Web mining, Web personalization and Recommender Systems.

Dr. J. B. Patil, is Principal at R. C. Patel Institute of Technology, Shirpur. He is a recognized Ph. D. guide in the subjects of Computer Engineering and Electronics and Telecommunication Engineering of North Maharashtra University, Jalgaon as well as NMIMS University, Mumbai. He has worked as Dean, Faculty of Engineering & Technology, Member of Academic Council and Member of Senate of North Maharashtra University, Jalgaon. He has also worked as Chairman, Board of Studies in Computer Engineering and Information Technology of North Maharashtra University, Jalgaon from January 2008 onwards. He is working as member of academic council at Dr. BabasahebAmbedkar Technological University, Lonere.