

Context-Based Rating Prediction using Collaborative Filtering and Linked Open Data

Vineet Kumar Sejwal
Department of Computer Science
Jamia Millia Islamia, New Delhi, India
vineetsejwal.jmi@gmail.com

Muhammad Abulaish, *SMIEEE**
Department of Computer Science
South Asian University, New Delhi, India
abulaish@ieee.org

ABSTRACT

Linked Open Data (LOD) consists of various knowledgebases, such as DBpedia, Yago, and Freebase, and uses structured data as features to conceptualize a domain of interest. During last few years, many researchers have shown how LOD can be utilized in various applications, including recommender systems that are used to map items and users generally on the basis of interest and similarity parameters. However, contextual features play an important role to improve the effectiveness of the recommender systems. In this paper, we propose a contextual feature-based rating prediction and recommendation technique using item-based collaborative filtering and LOD. To this end, we have generated a RDF graph representing items and their contextual features, which help to determine context-based similar items for recommendation using graph matching techniques. In order to extract contextual features for item profiling, we have used LOD and two famous movie data sources, Rotten Tomatoes and IMDB. We also propose a rating prediction model to predict the rating of the non-rated items with the help of the RDF graph and item-based collaborative filtering. The proposed approach is evaluated using *mean absolute error* and *root mean square error*, and performs significantly better in comparison to some of the standard baseline methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Human centered computing** → *Collaborative Filtering*;

KEYWORDS

Recommender System, Linked Open Data, Collaborative Filtering, Context-Based Similarity, RDF Graph

ACM Reference format:

Vineet Kumar Sejwal and Muhammad Abulaish, *SMIEEE*. 2019. Context-Based Rating Prediction using Collaborative Filtering and Linked Open Data. In *Proceedings of 9th International Conference on Web Intelligence, Mining and Semantics, Seoul, Republic of Korea, June 26–28, 2019 (WIMS2019)*, 9 pages. DOI: 10.1145/3326467.3326489

* Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WIMS2019, Seoul, Republic of Korea

© 2019 ACM. 978-1-4503-6190-3/19/06...\$15.00
DOI: 10.1145/3326467.3326489

1 INTRODUCTION

Recommender system generally recommends items based on content similarity and collaborative filtering, ignoring the *conditional usage* of the items and services by the users. The *conditional usage* of items means determining the fact that what products are used by which users and when, while making a recommendation. For example, people usually prefer to watch different types of movie along with different companion at different places. Such *conditional usage* constraints can be satisfied by the traditional recommender systems through incorporating context into them. Moreover, integration of contextual information, such as demographic location, time, companion, etc. into the traditional recommender systems may improve their accuracy. As defined in [1], "*context is any information that can be used to characterize the situation of an entity, such as person, place or object which is relevant in the interaction between the entity and an application, including the user and applications themselves*". Lieberman et al. [13] defined context as: "*context can be considered to be everything that affects computation, except the explicit input and output*". Similarly, Cantador and Castells [6] defined context as – "*the background topics under which activities of a user occur within a given unit of time*".

Structured data are published and interlinked using Linked Open Data (LOD), a web technology at the top of RDF (Resource Description Framework) and URI (Uniform Resource Identifier) for building hybrid knowledgebases. Like URL, URI is used to identify resources in name spaces, whereas RDF is used to represent information components in the form of a triplet $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ that can be easily interpreted by the machines. Figure 1 represents a sampler LOD cloud containing numerous interlinked knowledgebases. The LOD is publicly available in structured (RDF) format and can be consumed easily by humans or machines. DBpedia is the nucleus of LOD which is linked to almost every knowledgebase. Each resource in a knowledgebase has an external link which can be used to open the underlying resource in the linked knowledgebase. At present, approximately 150 billion RDF triples and 10000 linked datasets are available in LOD. The availability of such huge amount of information enables the applications of LOD in various fields, including the development of context-based recommender systems.

In this paper, we propose a contextual feature-based recommendation technique for movie domain using item-based collaborative filtering and LOD. The proposed technique aims to generate item profiles through modelling data from various sources into a RDF graph, wherein nodes represent the items and their contextual features, and edges represent the linkages between them. The RDF graph is mainly used to identify context-based similar items using graph matching techniques. For experiment, we have used data from DBpedia and two movie data sources (Rotten Tomatoes and

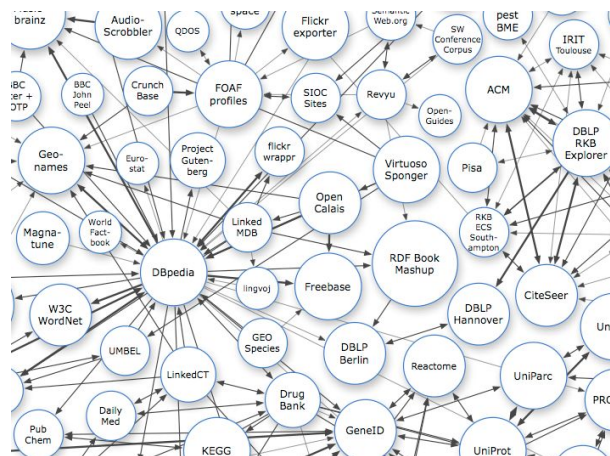


Figure 1: A sampler linked open data cloud

IMDB) and analyzed them using various information extraction and NLP techniques to identify various contextual features, such as *cast*, *director*, *topic*, *genre*, *sub-genre*, and *topics* for movie profiling. We also propose a rating prediction model to predict the rating of the non-rated items with the help of the RDF graph and item-based collaborative filtering. The proposed approach can be used to overcome some of the open issues, such as *black-box recommendation* and *cold-start* problem associated with the existing recommender systems. The proposed context-based recommendation technique is evaluated using standard performance evaluation metrics *mean absolute error* (MAE) and *root mean square error* (RMSE), and compared with some of the standard baseline methods.

The rest of the paper is organized as follows. Section 2 presents a brief overview and survey of the state-of-the-art recommendation methods. Section 3 presents the preliminary concepts. Section 4 describes the proposed context-based semantic similarity approach for rating prediction and recommendation, including labeled data graph generation and context-based semantic similarity calculation. Section 5 presents details about context-based rating prediction using item-based collaborative filtering. Section 6 presents the experimental setup and evaluation results. Finally, section 7 concludes the paper with future directions of work.

2 LITERATURE REVIEW

Recommender system uses users' attributes, such as profiles, tastes, history, preferences, and their interaction data for recommendation, and a traditional RS is generally represented as $R : user \times item \rightarrow rating$. Accordingly, various recommendation techniques, such as content-based approaches, collaborative filtering (CF), trust-aware recommendation, and hybrid approaches have been developed by the researchers. In order to improve the accuracy of a RS, Adomavicius et al. [2] introduced another dimension, context, which enhanced RS as $R : user \times item \times context \rightarrow rating$ to define context-aware recommender system (CARS). The CARS incorporates context information, such as time, companion, place etc. both for users and items to get the exact requirements of the users and their item preferences. In order to incorporate context in existing

RSs, three approaches – pre-filtering of contextual features, post-filtering of contextual features, and contextual modeling exist in literature [2]. Many researchers have proposed context-based item profiling to find the most similar items. Hawalah and Fasli [10] proposed a heuristic approach for item profiling in movie domain. They used movie features like genre, director, cast as item-related contextual features, and activity, time, location as user-related contextual features. Cosine similarity is used for finding the most similar items. Otebolaku and Andrade [17] proposed a case-based reasoning approach for item profiling in movie, music, and news domains, where time, location, activity, and artificial (environment) are the contextual dimensions. Pre-filtering paradigm is used with cosine similarity to find similar items. In [8], an item-item based model is proposed that works on finding both local and global item-item similar items based on user contexts for switching in different subsets. The authors in [19] proposed an approach for recommendation for groups of users. Here contextual influence of users are used for deriving groups by aggregating user preferences on certain contexts. Wao et al. [21] used user, item, and decision contexts for graph-based CARS in which genre, artist, etc. are considered as item contexts.

In recent years, many researches on recommender systems have used knowledge graphs and LOD [7, 14]. As a pre-requisite for the success of the semantic web, RDF was developed to represent related information components about resources as a triple $\langle subject, predicate, object \rangle$, where *subject* and *object* correspond to nodes, and *predicate* corresponds to a labeled edge connecting respective nodes in the RDF graph. Fouss et al. [9] developed a method for finding similarities between nodes of a graph for the development of a movie recommender system. Their method is based on random walk using Markov chain model for weighted graph, which contains relations such as “has_watched” and “belongs_to” for connecting people to movies and movie category nodes. Noia et al. [15] proposed a recommendation approach using content-based algorithm and LOD, wherein vector space model and cosine similarity are used for item profiling and similarity finding. Ostuni et al. [16] developed Cinemappy application for computing a context-based movie recommender system, where spatial and temporal values of users are used as contextual features, and SPARQL queries are used to extract information about movies from LOD.

A number of approaches for finding node similarity using graphs and sub-graphs are presented in [18][5][4]. The methods used in these approaches are max (min) common sub-graph (super graph), graph isomorphism, and iterative. In this work, we have used iterative approach for finding node similarity using similar neighborhoods. In iterative methods, initial similarity score between two graphs (nodes) is repeatedly refined using some update rules, such that $[sim_{(i,j)}]^{k+1} \leftarrow f(sim_{(i,j)}^k)$. In past, researchers have proposed various similarity measures for finding similarities between the items. Huang in [12] compared most of the similarity measures and distance functions to analyze their effectiveness in text documents clustering. Methods like Euclidean distance, averaged Kullback-Leibler divergence, Jaccard coefficient, Cosine similarity, and Pearson correlation coefficient have also been used to cluster text documents based on their similarity scores.

3 PRELIMINARIES

This section presents a brief description of some basic concepts, such as item context, linked open data, and Latent Dirichlet Allocation (LDA) that are used in our proposed method. Following sub-sections presents a detailed descriptions of these concepts.

3.1 Item Context

Item contextual features represent different possible contexts and conditions of an item for its consumption by the users. For example, in music domain, genre and artist can be used as context; in movie domain, genre, cast, and director can be the possible contextual features. In this paper, contexts for an item is defined as I_{j,c_k} , where $j = 1, 2, \dots, n$ are the n items and $\{c_1, c_2, \dots, c_k\}$ represent the context of item j . Each item can have multiple contexts and a particular context can have a number of values, called attributes. For example, an item I_1 having two contextual features, c_1 representing genre as context and horror as context value, whereas c_2 having sub-genre as context and dystopia as context value, can be represented as $I_{1(c_1, c_2)} = \{\langle genre : horror \rangle, \langle sub-genre : dystopia \rangle\}$.

3.2 Linked Open Data

Linked Open Data (LOD) can be defined as any structured web-based information that are connected using the technologies like HTTP and URI, based on the RDF standard for the semantic net. It can be considered as a globally accessible virtual cloud of linked data where anybody can access any data based on their authorization and also add any data without manipulating the original data source.

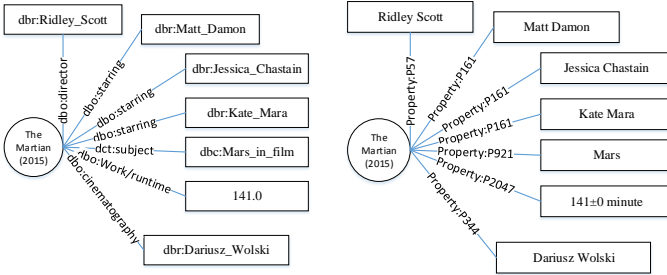


Figure 2: DBpedia and Wikidata representation of “the Martian” movie

For example, figure 2 presents a representation of “the Martian” movie and its contextual features from two knowledgebases – DBpedia and Wikidata. In DBpedia, features are extracted by using dbo, dct, and dbp, whereas in Wikidata there are certain property codes (e.g., P161, P921, P344, P2047) for the predicate edges that can be used for context extraction. LOD is a collection of all such knowledgebases under a common umbrella such that all information and features of a particular domain can be linked together.

SPARQL end-point can be used to gather information from RDF triples and URIs using the property names. For example, in figure 2, both DBpedia and Wikidata property codes are mentioned, and “[http://dbpedia.org/page/The_Martian_\(film\)](http://dbpedia.org/page/The_Martian_(film))” and “<https://www.wikidata.org/wiki/Q18547944>” are the corresponding URI for “the Martian” movie. Mapping is an important and mandatory step to

access data from LOD cloud. We applied a mapping procedure to get the URIs of all movies belonging to our dataset. Thereafter, we run SPARQL queries over LOD cloud to extract different LOD features that are later used as contextual features for the corresponding items. By using $\langle property, value \rangle$ representation, features and their values extracted from LOD cloud are stored in a structured format.

3.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a statistical modeling technique based-on generative probability model, primarily constrained with topic modeling in text documents [3]. The assumptions behind LDA are that documents with similar topics use similar groups of terms. Accordingly, documents can be represented as document-term matrix which can be factorized into two other matrices namely document-topic and topic-term matrix.

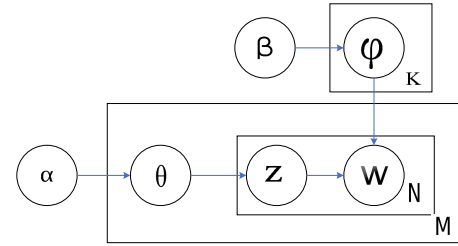


Figure 3: Graphical representation of the basic LDA model

Figure 3 presents the work-flow through graphical representation of the LDA model, where first box (outer one) represents the documents and second box (inner one) shows the repeated topics and words in the documents. M and N represent the documents and words inside documents. α and β are two parameters, representing document-topics distribution and topic-words distribution. θ_M and ϕ_K are the topics and words distributions, respectively for documents M and topics K .

Equation (1) computes the joint probability distribution for w words, z topics, and θ topic-word distribution using N number of topics and words as given in [3]. The equation (1) is further integrated using topic distribution θ and summation of z topics to find the marginal distribution of documents represented in equation (2). Finally, the corpus probability is obtained by taking the product of the sum of equation (2) for m documents.

$$p(\theta, z, w | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta) \quad (1)$$

$$p(w | \alpha, \beta) = \int (p(\theta | \alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n | \theta) p(w_n | z_n, \beta) \right) d\theta \quad (2)$$

LDA provides three levels of representations; first is corpus-level representation given by α and β , second is document-level representation given by θ_d , and third is word-level representation given by z_{dn} and w_{dn} . The first level is significant for corpus generation, second level works for document part, and third level is used for each word in each documents.

Table 1 presents a partial list of topics extracted using LDA over a corpus of review documents related to three well-known movies – *the Martian*, *San Andreas*, and *Gravity*.

Table 1: Topics for movies “the Martian”, “San Andreas” and “Gravity” using LDA

Movie name	Topics
The Martian	mars, space, survive, nasa, rescue, survival, solitude, martian, astronaut, science, visual, effects
San Andreas	rescue, earthquake, tsunami, directs, daughter, father, disaster, save, effects, visual
Gravity	space, gravity, earth, astronaut, survive, solitude, effects, visual, nasa

4 PROPOSED APPROACH

This section presents a detailed description of the proposed approach for contextual features based recommendation technique using item-based collaborative filtering and LOD. As shown in figure 4, it is implemented as a labeled directed graph using RDF. It also presents details about labeled directed graph generation and semantic similarity measures for rating prediction. In a labeled directed graph, entities (resources) are the nodes and labeled links are the edges representing relations between the entities. In other words, each link (labeled edge) connecting a pair of nodes represents a triplet $\langle \text{subject}, \text{predicate}, \text{object} \rangle$, where subject/object corresponds to the nodes and predicate corresponds to the label of the edge connecting the nodes. Further details about labeled directed graph generation and semantic similarity measures are presented in the following sub-sections.

4.1 Labeled Data Graph Generation

As discussed in the previous section, LOD is generally implemented as a labeled directed graph using RDF (*aka* RDF graph). The direction of edges are used to determine the *in-neighbor* and *out-neighbor* of the nodes. A RDF graph is a collection of triplets like $\langle \text{subject}, \text{predicate}, \text{object} \rangle$, where *subject* and *object* in a RDF triple are considered as nodes, and *predicate* (representing the relationship between subject and object) is considered as the label of the edge connecting the respective nodes. In a RDF graph, URIs (Uniform Resource Identifiers) are used to uniquely identify subjects, objects and predicates. Figure 5 presents a RDF graph in which subjects and objects are represented using rectangles and ovals, respectively, and predicates are represented using dashed labeled edges. A RDF graph can be used to determine similarity between two resources if either (i) the resources are directly related, or (ii) the resources are the subject of the triplets sharing the same predicates with same objects, or (iii) the resources are objects of the triplets sharing the same predicates with same subjects [16].

Linked open data can be defined as a 3-tuple $\langle V, P, S \rangle$, where $V = \{v_1, v_2, \dots, v_n\}$ represents the set of subject or object resources (nodes), $P = \{p_1, p_2, \dots, p_m\}$ represents the set of predicates (links), and $S = \{s_1, s_2, \dots, s_t\}$ is the set of statements (triplets) linking predicates with their respective subjects and objects. In a LOD

triplet, both source and destination of a directed edge can be either subject or object.

4.2 Context-Based Semantic Similarity

In this section, we present a formulation of Context-Based Semantic Similarity (CBSS) measure, which is used to calculate the similarity between items considering their contextual features.

The proposed CBSS measure is defined as a recursive process for calculating similarity between a pair of item nodes, say (i, j) . In order to initialize the process, an initial similarity score between each pair of nodes is calculated using equations (3) to (5).

$$\text{inSim}^1(i, j) = \frac{\min\{deg_{in}(i), deg_{in}(j)\}}{\max\{deg_{in}(i), deg_{in}(j)\}} \quad (3)$$

and

$$\text{outSim}^1(i, j) = \frac{\min\{deg_{out}(i), deg_{out}(j)\}}{\max\{deg_{out}(i), deg_{out}(j)\}} \quad (4)$$

$$\text{CBSS}^1(i, j) = \frac{\text{inSim}^1(i, j) + \text{outSim}^1(i, j)}{2} \quad (5)$$

Thereafter, a recursive process is applied to the enumeration functions containing the neighbors of both nodes to find similar *in-neighbors* and *out-neighbors*. Equation (6) defines the recursive process starting from $k = 1$, using two similarity functions $\text{inSim}^k(i, j)$ and $\text{outSim}^k(i, j)$ that are defined in equations (7) and (8), respectively. These functions are mainly used to compute similarity between the contextual features of nodes i and j . In these functions, $d_{in}(i)$ and $d_{out}(i)$ are used to represent the in-degree and out-degree of node i ; whereas, other notations are self-explanatory.

$$\text{CBSS}^{k+1}(i, j) \leftarrow \frac{\text{inSim}^{k+1}(i, j) + \text{outSim}^{k+1}(i, j)}{2} \quad (6)$$

$$\text{inSim}^{k+1}(i, j) = \text{CBSS}^k(i, j) \frac{1}{\max\{d_{in}(i), d_{in}(j)\}} \times \sum_{l=1}^{d_{in}(i)} \sum_{m=1}^{d_{in}(j)} \text{sim}(\text{inNeighbor}_l^{(i)}, \text{inNeighbor}_m^{(j)}) \quad (7)$$

$$\text{outSim}^{k+1}(i, j) = \text{CBSS}^k(i, j) \frac{1}{\max\{d_{out}(i), d_{out}(j)\}} \times \sum_{l=1}^{d_{out}(i)} \sum_{m=1}^{d_{out}(j)} \text{sim}(\text{outNeighbor}_l^{(i)}, \text{outNeighbor}_m^{(j)}) \quad (8)$$

The *sim* function in equations (7) and (8) is defined using equation (9) to determine the similarity between a pair of contextual feature (f_l, f_m) . If $f_l = f_m$. The similarity value of the features is calculated using a probability function $pr(f)$, which is defined using equation (10), otherwise it is set to 0. In equation (10), n is the number of entities (resources) with which the feature value is associated, and N is the total number of entities presents in the data graph. The $pr(f)$ represents the probability of finding the completeness value of the matching features in the data graph. In most of the existing state-of-the-art methods, matched and unmatched nodes are assigned the value 1 and 0, respectively, which is not able to differentiate between frequently and rarely occurring features.

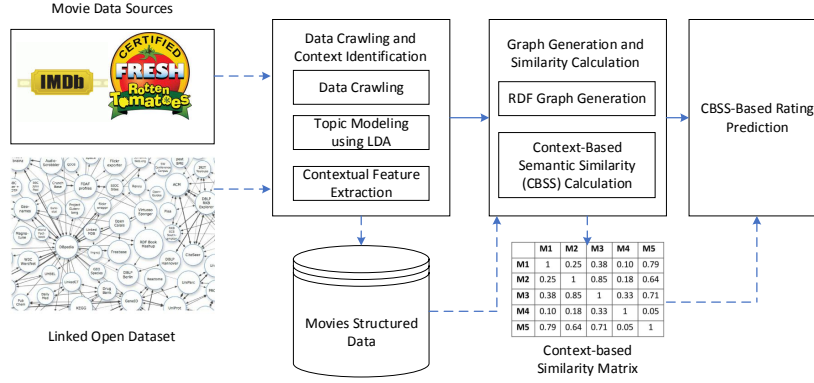


Figure 4: Work-flow of the proposed rating prediction and recommendation approach

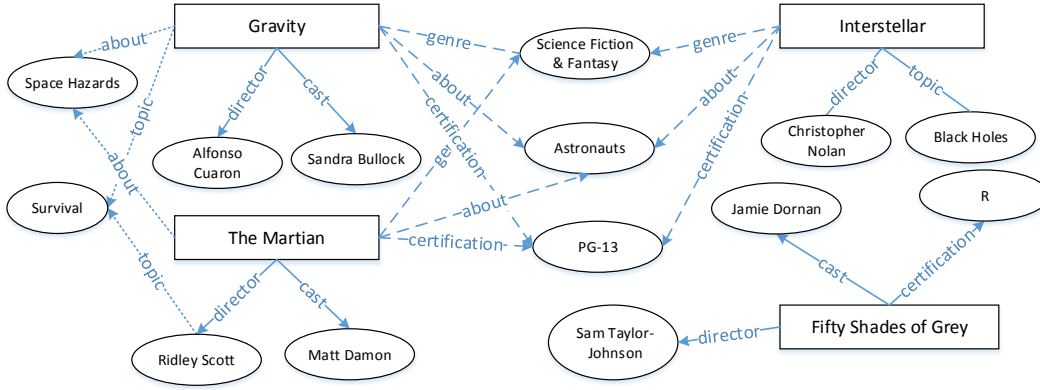


Figure 5: An exemplar RDF graph corresponding to movie LOD

The proposed similarity function assigns higher weight to rare occurring features in comparison to the frequent occurring features, as high probable features provide less amount of information in comparison to the distinctive features that are specific in amount and are highly informative.

$$\text{sim}(f_i, f_m) = \begin{cases} 1 - \text{pr}(f_i), & \text{if } f_i = f_m \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$\text{pr}(f) = \frac{n}{N} \quad (10)$$

The CBSS value between a pair of nodes is updated at each iteration and converges towards a similarity value between 0 and 1. The process is stopped when the difference between two successive CBSS values at $(k+1)$ th and k th iterations is less than or equal to the pre-defined threshold δ , as given in equation (11). Using the CBSS function defined earlier, we generate a similarity matrix M (equation (12)) to store the item-item similarity scores for rating prediction and recommendation purpose.

$$|CBSS^{k+1}(i, j) - CBSS^k(i, j)| \leq \delta \quad (11)$$

$$M[i, j] = \begin{cases} 1, & \text{if } i = j \\ CBSS(i, j), & \text{otherwise} \end{cases} \quad (12)$$

5 RATING PREDICTION

This section presents the rating prediction of an item for a particular user using context-based semantic similarity approach. To predict the rating of an item (say i) for a user (say u), top- k items similar to i are identified using the similarity matrix M . Thereafter, for each top- k items, the similarity values are combined with u 's ratings (if available) using the equation (13) given by [20] to predict the rating of u on i (\hat{r}_{ui}), where, I_u^n represents the top- n similar items to i rated by u , and r_{uj} is the actual rating value rated by user u for item j .

$$\hat{r}_{ui} = \frac{\sum_{j \in I_u^n} CBSS(i, j) r_{uj}}{\sum_{j \in I_u^n} |CBSS(i, j)|} \quad (13)$$

It is possible that some users give high or low rating on certain items due to biasness or critical nature, which may affect the rating prediction. To overcome such biasness issue, the authors in [20] proposed the incorporation of a first order approximation as given in equation (14), where $b_{ui} = \mu + b_u + b_i$, μ represents the overall

average rating of items in the dataset, b_i and b_u are deviations of user and item ratings with respect to average rating μ .

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in I_u^n} CBSS(i, j)(r_{uj} - b_{uj})}{\sum_{j \in I_u^n} |CBSS(i, j)|} \quad (14)$$

6 EXPERIMENTAL SETUP AND RESULTS

In this section, we discuss the experimental setup and results to establish the efficacy of the proposed context-based rating prediction and recommendation approach. As stated earlier, we have used data from two well-known data sources in movie domain – Rotten Tomatoes and IMDB, and two LOD sources – DBpedia and WikiData. The proposed approach is evaluated using standard evaluation metrics – MAE and RMSE. We have also presented a comparative analysis of the proposed approach with some of the baseline methods. Further details about these processes are given in the following sub-sections.

6.1 Dataset and Results

For experimental evaluation, we developed a crawler in Python language using some standard libraries to retrieve data, such as movie name, genre, cast, director, reviews, certification, from Rotten Tomatoes and IMDB data sources. We have also written SPARQL queries to retrieve data from DBpedia and WikiData that are not available at Rotten Tomatoes and IMDB. A brief description of different categories of data retrieved from movie data sources is given in table 2.

Table 2: Dataset description

Category	Freq. count	Source
Movies	1102	https://www.rottentomatoes.com
Reviewers	78356	http://www.imdb.com
Reviews	174576	http://www.imdb.com
Critics	5650	http://www.rottentomatoes.com
Ratings	160606	http://www.imdb.com

As shown in table 2, different categories of data such as users and critics reviews and users’ ratings, along with various contextual features, such as genre, sub-genre, certification, direction, casting, etc. were retrieved for a total number of 1102 movies. At IMDB, each movie is rated by the users on a 10-point scale, with 1 representing the lowest rating and 10 representing the highest rating. The rating values for each movie were also extracted and stored in a structured format. LDA was applied over the review documents to identify topics for characterizing the respective movies. The data retrieved from LOD knowledge sources were segmented using the phrases like “about” and “based-on” and stored in a structured format for characterizing the respective movies. Using all these information, an RDF graph was generated which consists of 109450 triples of the form $\langle \text{subject}, \text{predicate}, \text{object} \rangle$. Thereafter, we generated similarity matrix of order 1102×1102 using the context-based semantic similarity function defined in the previous section for rating prediction and recommendation purpose. Table 3 presents a comparative analysis in terms of similar movies with or without LDA for the movie “Inside Out”. LDA plays a very important part for finding

Table 3: Top-5 movies similar to “Inside Out” computed with and without LDA

Similarity with LDA		Similarity without LDA	
Movie	Sim. Value	Movie	Sim. Value
Planes	0.2248	Rio 2	0.109
The Peanut Movie	0.1824	Free Birds	0.107
Minions	0.1706	Minions	0.1064
The Good Dinosaur	0.1656	Turbo	0.1061
Rio 2	0.1558	Planes	0.0987

similar movies. In next section, we have shown the importance of LDA for rating prediction. Table 4 presents a partial result showing the predicated and actual ratings of 5 movie items by 10 users.

6.2 Evaluation Results

This section presents the evaluation results of the proposed context-based rating prediction and recommendation approach using standard metrics MAE and RMSE. The MAE is calculated as the mean of the absolute differences between the predicted and actual ratings [11], as defined in equation (15). On the other hand, RMSE is defined as the standard deviations of the prediction errors (residuals), as given in equation (16). It penalizes large error values and basically determines the intensity of data to a line of best fit. In these equations, \mathcal{T} is the test dataset containing movies and their ratings by different users, \hat{r}_{ui} is the predicted rating for user u on item i , and r_{ui} is the actual rating of item i given by user u . Table 5 presents MAE and RMSE values for different values of top- k similar items for our proposed CBSS-based rating prediction and recommendation method.

$$MAE = \frac{\sum_{(ui) \in \mathcal{T}} |\hat{r}_{ui} - r_{ui}|}{|\mathcal{T}|} \quad (15)$$

$$RMSE = \sqrt{\frac{\sum_{(ui) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}{|\mathcal{T}|}} \quad (16)$$

6.3 Comparative Analysis

In this section, we present a comparative analysis of our proposed CBSS-based rating prediction and recommendation method with three standard baseline methods, which are briefly described in the following paragraphs.

- **Baseline Predictor:** The base-line prediction method is used to deal with those items and users that suffer with cold-start problem, and it is defined as the sum of overall average ratings (μ), user deviation (b_u), and item deviation (b_i). Baseline predictor is defined as $(\mu + b_u + b_i)$.
- **Jaccard Similarity:** Jaccard similarity compares the elements of two sets to determine the degree of their overlap. It is defined as the ratio of the cardinality of the intersection of the sets to the cardinality of the union of the sets. Mathematically, the Jaccard similarity between two sets,

Table 4: A partial result showing predicted and actual ratings of users on different movies

Users	Movies (actual/predicted ratings)				
	Martian	Sicario	Maze Runner	Chappie	Mad-Max
djangozelf-12351	9/7.566	2/4.688	7/5.588	6/5.462	2/3.158
Charlie Picart	7/7.835	8/8.409	6/5.289	3/5.123	9/8.820
Michael Seng Wah	9/9.269	9/8.231	8/6.428	9/7.538	9/7.989
Bob Rutzel	7/8.499	9/7.401	7/5.987	7/8.258	9/7.402
AliceofX	7/6.266	7/6.810	5/5.420	8/9.623	9/7.304
Vivekmaru45	7/6.853	7/6.138	4/6.192	5/4.160	7/8.236
Joe	9/7.285	6/4.828	8/7.900	9/7.102	7/7.913
ZULFIQAR RAJA	8/6.289	9/8.564	8/9.158	8/7.022	9/9.582
Thanos Alfie	9/7.865	6/4.528	6/6.014	8/8.952	9/7.825
s3276169	7/5.825	4/5.635	5/7.128	5/6.158	9/7.766

Table 5: MAE and RMSE values of the proposed CBSS-based rating prediction and recommendation method

	k=10		k=20		k=30		k=40	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBSS_{LOD+LDA}	0.701	0.914	0.693	0.903	0.671	0.861	0.714	0.904

Table 6: A comparative analysis of the proposed CBSS-based rating prediction and recommendation method with existing baseline methods

	k=10		k=20		k=30		k=40	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Jaccard Similarity	1.308	1.746	0.883	1.196	0.876	1.179	0.891	1.186
Pearson Correlation	1.322	1.758	0.898	1.218	0.895	1.192	0.915	1.205
Base Line Prediction	1.285	1.621	0.947	1.26	0.906	1.208	0.919	1.204
CBSS_{LOD}	0.7314	0.9348	0.7208	0.9215	0.7014	0.9018	0.7445	0.9298
CBSS_{LOD+LDA}	0.701	0.914	0.693	0.903	0.671	0.861	0.714	0.904

say A and B , can be calculated using equation (17).

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (17)$$

- **Pearson Correlation Coefficient:** Pearson Correlation Coefficient (PCC) is used to find the relationships between two sets, i.e. to determine whether two sets are positively correlated, negatively correlated, or unrelated. Mathematically, PCC between two sets, say $X = \{x_1, x_2, \dots, x_n\}$

and $Y = \{y_1, y_2, \dots, y_n\}$, can be determined using equation (18).

$$PCC(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (18)$$

- **Context-based similarity without LDA:** In this comparative method similarity between two items, say i and j , is computed without considering the topics generated by LDA from item reviews. For this type of similarity, we only consider contextual features extracted from LOD and movie data sources. Table 3 presents top-5 similar movies

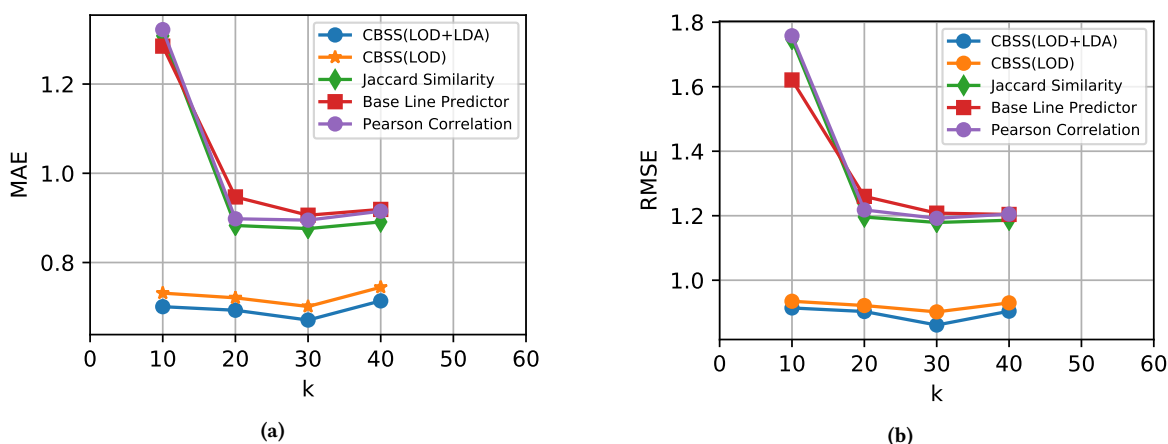


Figure 6: (a) MAE values of the proposed CBSS-based rating prediction and baseline methods at different k values (b) RMSE values of the proposed CBSS-based rating prediction and baseline methods at different k values

computed using LDA and without LDA for the movie “Inside Out”.

Table 6 and figure 6(a),6(b) presents the comparison results in terms of MAE and RMSE for different top- k values. It can be observed from this table that MAE and RMSE values for our proposed method is lowest in comparison to other standard baseline methods. The values of MAE and RMSE varied in the range of 0.671 to 0.714 and 0.861 to 0.914, respectively for proposed method, and both are minimum for $k = 30$, which is significantly better than the existing methods. Also, the values of MAE and RMSE for the proposed approach without LDA varied in the range of 0.7014 to 0.7445 and 0.9018 to 0.9348. The proposed approach including LOD and LDA performed 3.05% better in terms of MAE and 4.08% better in terms of RMSE at $k = 30$ in comparison to our proposed approach which only includes LOD. On analysis, it is found that in comparison to the baseline prediction, the proposed method performed 23.5% better in terms of MAE and 26% better in terms of RMSE at $k = 30$. Similarly, the proposed method performed 20.5% better in terms of MAE and 31.9% better in terms of RMSE in comparison to the Jaccard similarity method. Finally, the proposed method performed 22.5% better in terms of MAE and 33.1% better in terms of RMSE in comparison to the Pearson correlation similarity method.

7 CONCLUSION

In this paper, we have proposed a contextual features based rating prediction and recommendation approach using item-based collaborative filtering and linked open data. The proposed approach can be seen as an amalgamation of analyzing contents and structural data using labeled data graph for the development of effective recommender systems. Contents data can be processed using various IE and NLP techniques to identify features (contexts) characterizing the entities of interest, whereas structural data can be used to establish linkages between various entities and their features. One of the appealing advantages of using linked data graph to model both content and structural data is to handle cold-start problem using collaborative filtering approaches. Combining data from LOD

knowledgebases with movie data sources is helpful to define context in a broader perspective. In this paper, we have used item-based collaborative filtering, which mainly facilitate the rating prediction for new items. However, it can be extended to apply user-based collaborative filtering to facilitate recommendations for new users. Application of graph mining techniques to deal with huge, heterogeneous, and multi-dimensional graphs also seems one of the promising research areas in the field of recommender system.

REFERENCES

- [1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggle. 1999. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*. Springer, Berlin, Heidelberg, 304–307. https://doi.org/10.1007/3-540-48157-5_29
- [2] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alexander Tuzhilin. 2011. Context-Aware Recommender Systems. *Association for the Advancement of Artificial Intelligence*, 32, 3 (2011), 217–253. <https://doi.org/10.1609/aimag.v32i3.2364>
- [3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of machine Learning research* 3 (2003), 993–1022.
- [4] Vincent D. Blondel, Anahi Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. 2004. A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. *SIAM Rev.* 46, 4 (2004), 647–666. <https://doi.org/10.1137/S0036144502415960>
- [5] Horst Bunke. 1999. Error Correcting Graph Matching: On the Influence of the Underlying Cost Function. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 9 (1999), 917–922. <https://doi.org/10.1109/34.790431>
- [6] Iván Cantador and Pablo Castells. 2009. Semantic Contextualisation in a News Recommender System. In *Proceedings of 1st Workshop on Context-Aware Recommender System (CARS-RecSys’09)*. ACM Press, New York, NY, USA.
- [7] Rose Catherine and William Cohen. 2016. Personalized Recommendations using Knowledge Graphs: A Probabilistic Logic Programming Approach. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys’16)*. ACM Press, Boston, Massachusetts, USA, 325–332. <https://doi.org/10.1145/2959100.2959131>
- [8] Enbl Christakopoulou and G. Karypis. 2016. Local Item-Item Models for top-n Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys’16)*. ACM Press, Boston, Massachusetts, USA, 67–74. <https://doi.org/10.1145/2959100.2959185>
- [9] Francois Fouss, Alain Pirotte, and Marco Sareren. 2005. A Novel Way of Computing Similarities between Nodes of a Graph, with Application to Collaborative Recommendation. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI’05)*. IEEE, Compiegne, France, 550–556. <https://doi.org/10.1109/WI.2005.9>
- [10] Ahmad Hawalah and Maria Fasi. 2014. Utilizing Contextual Ontological User Profiles for Personalized Recommendations. *Expert Systems with Applications* 41, 10 (2014), 4777–4797. <https://doi.org/10.1016/j.eswa.2014.01.039>

- [11] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53. <https://doi.org/10.1145/963770.963772>
- [12] Anna Huang. 2008. Similarity Measures for Text Document Clustering. In *Proceedings of the 6th Conference of New Zealand Computer Science Research Student Conference (NZCSRSC' 08)*. NZCSRSC, Christchurch, New Zealand, 49–56.
- [13] Henry Lieberman and Ted Selker. 2000. Out of Context: Computer Systems that Adapt to, and Learn from, Context. *IBM Systems Journal*, 39, 3.4 (2000), 617–632. <https://doi.org/10.1147/sj.393.0617>
- [14] Tommaso D. Noia. 2016. Recommender Systems Meet Linked Open Data. In *Proceedings of the 16th International Conference on Web Engineering (ICWE' 16)*. Springer, Cham, Lugano, Switzerland, 620–623. https://doi.org/10.1007/978-3-319-38791-8_61
- [15] Tommaso D. Noia, Roberto Mirizzi, Vito C. Ostuni, Davide Romito, and Markus Zanker. 2012. Linked Open Data to Support Content-based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS' 12)*. ACM Press, Graz, Austria, 1–8. <https://doi.org/10.1145/2362499.2362501>
- [16] Vito C. Ostuni, Tommaso D. Noia, Roberto Mirizzi, Davide Romito, and Eugenio D. Sciascio. 2012. Cinemappy: A Context-Aware Mobile App for Movie Recommendations Boosted by DBpedia. In *Proceedings of the 1st International Workshop on Semantic Technologies meet Recommender Systems and Big Data (CEUR–SeRSy' 12)*, Vol. 919. CEUR, Boston, USA, 25–36.
- [17] Abayomi M. Otebolaku and Maria T. Andrade. 2015. Context-aware Media Recommendations for Smart Devices. *Journal of Ambient Intelligence and Humanized Computing* 6, 1 (2015), 13–36. <https://doi.org/10.1007/s12652-014-0234-y>
- [18] Marcello Pelillo. 1998. Replicator Equations, Maximal Cliques, and Graph Isomorphism. In *Proceedings of the 11th International Conference on Neural Information Processing Systems (NIPS' 98)*. MIT Press Cambridge, USA, Denver, Colorado, USA, 550–556.
- [19] Elisa Quintarelli, Emanuele Rabosio, and Letizia Tanca. 2016. Recommending New Items to Ephemeral Groups Using Contextual User Influence. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys' 16)*. ACM Press, Boston, Massachusetts, USA, 285–292. <https://doi.org/10.1145/2959100.2959137>
- [20] J.B. Schafer, Dan Frankowski, Jonathan L. Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. *The Adaptive Web* 4321 (2007), 291–324. https://doi.org/10.1007/978-3-540-72079-9_9
- [21] Weilong Yao, Jing He, Guangyan Huang, Jie Cao, and Yanchun Zhang. 2015. A Graph-based Model for Context-aware Recommendation Using Implicit Feedback Data. *World Wide Web* 18, 5 (2015), 1351–1371.