Final version of the accepted paper. Cite as: *"Vineet K. Sejwal and Muhammad Abulaish, A Hybrid Recommendation Technique using Topic Embedding for Rating Prediction and to Handle Cold-Start Problem, Expert Systems with Applications, Volume 209, 15 December 2022, pp. 118307. DOI: https://doi.org/10.1016/j.eswa.2022.118307 "*

A Hybrid Recommendation Technique using Topic Embedding for Rating Prediction and to Handle Cold-Start Problem

Vineet K. Sejwal[a] (vineetsejwal.jmi@gmail.com), Muhammad Abulaish, *SMIEEE*[b] (abulaish@sau.ac.in)

[a] Department of Computer Science, Jamia Millia Islamia, New Delhi, India

[b] Department of Computer Science, South Asian University, New Delhi, India

Corresponding Author:

Muhammad Abulaish, *SMIEEE*

Department of Computer Science, South Asian University, New Delhi, India

Email: abulaish@sau.ac.in

Abstract

Recommender systems aim to estimate item ratings and recommend items based on the users' interests. The traditional recommender systems generally consider user-item rating information for rating prediction, but they suffer from various limitations, such as *data sparsity*, *black-box recommendation*, and *cold-start* problems. As a result, researchers have proposed amalgamating contextual information with rating data to provide effective recommendations. Although user-generated data in the form of reviews are a rich source of contextual information, they are rarely utilized in recommender algorithms. This study presents a hybrid recommendation technique, called `RecTE`, using rating data and topic embedding, which is an amalgamation of word embedding and topic modeling techniques. The novelty of `RecTE` lies in predicting item ratings using topic embeddings learned by incorporating local and global contextual information and integrating them with user-based collaborative filtering. `RecTE` is empirically evaluated over three real-world datasets – `YelpNYC`, `YelpZip` and `TripAdvisor`. This technique performs significantly better in comparison to nine baselines and five state-of-the-art recommendation techniques. On empirical analysis, we found that incorporating topic embedding in `RecTE` makes it capable of performing significantly better and handle *cold-start* problems effectively in comparison to the existing recommendation approaches.

Keywords: Collaborative Filtering, Topic Modeling, Word Embedding, Topic Embedding, Cold-Start

1. Introduction

With the development of Web 2.0 and the explosive growth of the Internet, abundant information is available on the Web. Hundreds of thousands of e-commerce merchants sell and purchase countless products, such as movies, music, books, grocery items, electronic gadgets, hotels, and tourist destinations.

Each one of these products generates a considerable amount of information, such as product details, user profiles and their reviews, causing the *information overload* problem. The *information overload* problem demands a personalization system to recommend items based on the users' requirements and interests (Sejwal & Abulaish, 2021). Recommender systems play a vital role in information filtering and they are mainly used by the e-commerce organizations. A traditional recommender system is generally considered as a 2-dimensional mapping function $\mathcal{R} : user \times item \rightarrow rating$, which predicts ratings based on the historical ratings of the similar users. However, this system suffers from various limitations, such as *data sparsity*, *black-box recommendation*, and *cold-start* problems that degrade the recommendation efficacy (Sejwal et al., 2020).

To deal with the aforementioned issues, many researchers have used embedded textual information within user reviews for rating prediction. The user reviews are generally categorized as *specific reviews* and *generic reviews* (Bauman & Tuzhilin, 2014). In specific reviews, users express their personal experience on different aspects of the items, such as food, room quality, location, gym, and room service. From another aspect, generic reviews represent the overall impression of the users about the items. On analysis, specific reviews provide more contextual information, and they can be used for efficient user and item profiling.

Similarly, a text corpus contains local and global contextual information. The local context models a language from a local viewpoint in such a way that the semantically similar words are closer to each other. Local contextual information can be extracted using word embedding models, such as `Word2Vec` (Mikolov et al., 2013) and Neural Probabilistic Language Model (NPLM) (Bengio et al., 2003). The word embedding models are used to learn linguistic and semantic regularities in a text corpus. The assumption behind the word embedding model is based on the fact that co-occurring words are contextually related to each other. In addition, they exhibit similar semantic properties in the corpus. In the existing literature, researchers have effectively used the local contextual models for user and item modeling in recommender systems, such as venue recommendation (Manotumruksa et al., 2016) and music recommendation (Cheng et al., 2017). From another aspect, the global context represents the overall thematic orientation of a corpus modeled using various topic modeling techniques, such as Non-negative Matrix Factorization (NMF), Latent Dirichlet Allocation (LDA) (Blei et al., 2003), and Latent Semantic Indexing (LSI) (Liu et al., 2016). For global context, modeling of documents and topics is done in the form of topics distribution and multinomial words distribution, respectively. Similar to the local context, the global context is also used for user and item profiling to further enhance the recommendation accuracy. Although local and global contexts improve the rating prediction accuracy, both have certain limitations. The local context captures only local contextual information because they only consider words that are semantically and syntactically related and ignore those words that contribute globally. The local context is also affected by window size and review length because a small window ignores far away co-occurring words, whereas a long review hampers the information extraction efficiency. From another aspect, global context considers the overall contributions of the words toward topic generation, ignoring the semantic association between the words.

In the existing literature, many approaches exist for designing recommender systems using either local or global contextual information. However, to the best of our knowledge, no study has utilized both local and global contexts extracted from the specific and generic reviews for recommendation. We believe that incorporating both contexts will find more similar users, thereby improving the accuracy of the recommendation systems. Accordingly, this study presents a hybrid recommendation technique, called `RecTE`, which predicts ratings using a user-based collaborative model based on rating data and topic embedding, which exploits local and global contexts. The motive behind incorporating both contexts is to learn better word embeddings and improve topic discovery (Huang et al., 2008). `RecTE` is empirically evaluated on three publicly available datasets – `YelpNYC`, `YelpZip`, and `TripAdvisor`, containing user reviews and other related information using various error-based, decision support-based, and rank-based evaluation metrics. Incorporating topic embedding, which is an amalgamation of word embedding and topic modeling, improves the overall performance of RecTE. On empirical analysis, RecTE shows better performance in comparison to the standard recommendation approaches to handle the *cold-start* problem.

In short, the main contributions of this study can be summarized as follows:

- Presenting a novel recommendation approach, RecTE, which uses rating data and topic embedding and incorporates them into user-based collaborative filtering for effective rating prediction and handle cold-start problem.
- Empirically assesses the impact of topic embedding on recommendation, in comparison to the word-level and document-level embeddings.
- Empirically evaluates the proposed recommendation approach over multiple real-world datasets and presents a comparative analysis with several state-of-the-art and baseline approaches.
- Empirically assesses the effectiveness of the proposed recommendation approach to deal with the *cold-start* problem.

The rest of the paper is organized as follows. Section 2 presents a concise review of the word embedding and topic modeling-based recommendation techniques. Section 3 presents the functional details of the proposed RecTE, including local and global contexts learning, collaborative model design, and rating prediction using user-based collaborative filtering. Section 4 presents the experimental set up and evaluation results. Finally, section 5 concludes the study with future directions of research.

2. Related Work

This section presents a review of the existing literature on recommender systems. As one of the unique features of RecTE lies in the use of topic embedding, which is an amalgamation of word embedding and topic modeling, we have divided the literature survey into two parts, namely, word embedding-based recommender systems and topic modeling-based recommender systems.

Notation	**Description**
D	Document-term matrix
K	Number of topics
E	Dimension of the embedding space
N	Number of documents
U	Document-topic matrix
U'	Topic-term matrix
V	Number of terms
M	Word co-occurrence matrix
W	Word embedding matrix
W'	Context word embedding matrix
T	Topic embedding matrix
t_k	k^{th} topic embedding
u'_v	v^{th} word topic distribution
w'_v	v^{th} word context embedding

Table 1: Basic notations and their descriptions

2.1. Word Embedding-Based Recommender Systems

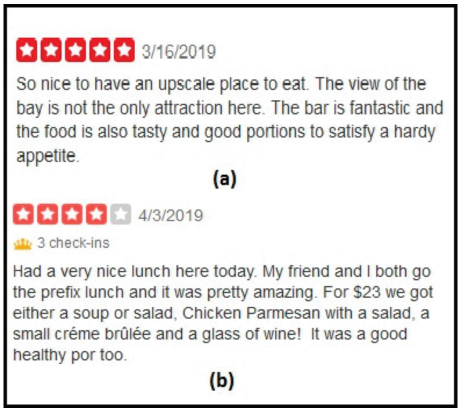
Word embeddings are used to represent words in a low-dimensional vector space by learning semantic regularities and linguistic information by analyzing the local word co-occurrence information in a large text corpus. In natural language processing, word embeddings were introduced as a NPLM (Bengio et al., 2003), and later on, many approaches were proposed to improve its efficacy (Collobert et al., 2011). Word embeddings can be categorized as frequency-based and prediction-based embeddings. In frequency-based embeddings, the frequency of the words is used to find the semantic similarities, such as word co-occurrence and tf-idf. From another aspect, Continuous Bag of Words (CBOW) and *skip-gram* models (Mikolov et al., 2013) are based on the context word and word prediction methods. Popular techniques, such as `Word2Vec` and LSI, (Deerwester et al., 1990) are frequently used for converting text corpora into word vectors. Recently, Matrix Factorization (MF) methods are gaining more recognition in recommender systems because MF is capable of handling large datasets using low-rank approximation techniques. As word embeddings also use low-dimensional vector space representation for the words, incorporation of word embeddings into the existing recommender systems can further improve their prediction accuracy. However, recommender systems using MF models suffer from the data sparsity problem, wherein users rate very few items. To handle this issue, authors in (Manotumruksa et al., 2016) introduced a regularized MF model for venue recommendation that incorporates social network information and word embeddings learned from review documents for calculating the similarity among the users. Musto et al. (Musto et al., 2016) introduced a Content-Based Recommender System (CBRS), wherein word embeddings are learned from the Wikipedia data. The learned word embeddings are later used to design various baseline techniques for comparative analysis. In continuation, they proposed a Deep Content-Based Recommender System (DeepCBRS) (Musto et al., 2018) that learns effective representations of items based on the textual features identified from Linked Open Data (LOD). In addition to using textual data, in some studies, word embeddings are constructed from non-textual data for developing rating prediction and recommendation systems. Ozsoy (Ozsoy, 2016) used word embeddings generated from non-textual features, such as check-in and check-out data of the users for hotel recommendations. Sheu and Li (Sheu & Li, 2020) proposed a Context-Aware Graph Embedding (CAGE) model that constructs a knowledge graph to extract semantic-level information for a news-based recommender system. Liu et al. (Liu et al., 2019) proposed a Dual Attention Mutual Learning (DAML) recommender system that used both ratings and reviews for recommendation. DAML uses local and mutual attention to learn features from reviews, that are integrated with rating features for item prediction. Liu et al. (Liu et al., 2021) introduced a multi-task recommendation model based on dual attention, which integrates user ratings, reviews, and review helpfulness votes for rating prediction.

2.2. Topic Modeling-Based Recommender Systems

The recommender system uses user preferences, profiles, tastes, history, and interactions to recommend the most relevant items. Generally, a two-dimensional recommender system is defined as $\mathcal{R} : user \times item \rightarrow rating$. Topic modeling is generally used to generate topics from a collection of documents. The assumption behind topic modeling is that each text document and each topic are a mixture of topics and distribution of words, respectively. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and Latent Semantic Indexing (LSI) are the two most popular techniques for generating topics from text documents. The MF approach is commonly used in topic modeling and factorizes document-term matrix into two lower dimensional matrices. One of the most common algorithms used in MF is the Stochastic Gradient Descent (SGD), which incorporates topic models to extract latent features (Tan et al., 2016). Jing et al. (Jing et al., 2015) used Laplace distribution for the factorization of matrices that are later used for topic generation. Then, Chen et al. (Chen et al., 2014) introduced a collaborative model that used topic modeling to mine user and item contents. Hu and Ester (Hu & Ester, 2013) presented a location-based recommender system, wherein they used topic modeling to find textual and spatial aspects of the users for predicting their locations. Wang and Blei (Wang & Blei, 2011) introduced collaborative filtering-based recommender system that recommends scientific articles to users using probabilistic topic modeling. The proposed approach in (Wang & Blei, 2011) used users' and items' latent features to recommend existing and upcoming scientific articles. Zhang et al. (Zhang et al., 2019) introduced a Dynamic Attention Integrated Neural Network (DAINN) model for news-based recommendations. DAINN jointly exploits users main purpose in the current session, user behavior sequence patterns, and users long-term interests to find their tastes. Topic modeling is used to build topic space for users to compute their long-term interests. Pena et al. (Pena et al., 2020) used textual reviews and topic modeling to generate topic space, which is used to construct user and item embeddings for recommendations.

3. Proposed Recommendation Approach

This section presents the functioning details of our proposed recommendation approach, `RecTE`, which consists of four different functionalities – (i) review categorization, (ii) local and global contextual information extraction, (iii) topic embedding learning, and (iv) user similarity computing and rating prediction. Further details about these functionalities are presented in the following subsections.



★★★★★ 3/16/2019

So nice to have an upscale place to eat. The view of the bay is not the only attraction here. The bar is fantastic and the food is also tasty and good portions to satisfy a hardy appetite.

(a)

★★★★☆ 4/3/2019

3 check-ins

Had a very nice lunch here today. My friend and I both go the prefix lunch and it was pretty amazing. For $23 we got either a soup or salad, Chicken Parmesan with a salad, a small créme brûlée and a glass of wine! It was a good healthy por too.

(b)

Figure 1: An exemplar (a) generic review, and (b) specific review

3.1. Review Categorization

This section presents the process of categorizing reviews into two classes – specific and generic reviews. Specific reviews describe the particular experience of the users containing detailed information about their visits, e.g., visiting a hotel, restaurant, or a tourist place. From another aspect, generic reviews present a general overview and an overall impression of the users. Figure 1 shows an example of specific and generic review. From these reviews, the specific review presents a rich set of contextual information about various entities,

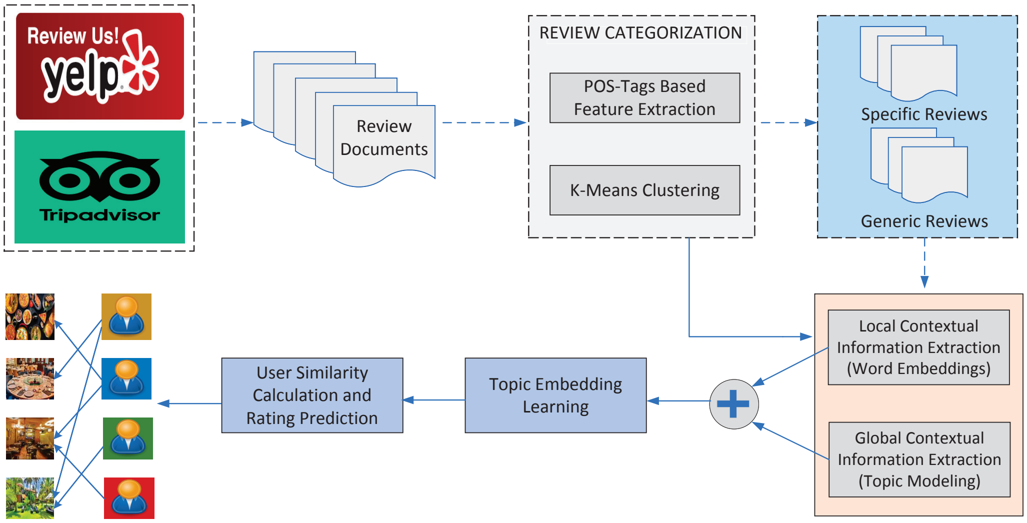

Review Us!
yelp
Tripadvisor
Review Documents
REVIEW CATEGORIZATION
POS-Tags Based Feature Extraction
K-Means Clustering
Specific Reviews
Generic Reviews
Local Contextual Information Extraction (Word Embeddings)
Global Contextual Information Extraction (Topic Modeling)
Topic Embedding Learning
User Similarity Calculation and Rating Prediction

Figure 2: Work-flow of the proposed `RecTE` system

such as companion, meal-time, and services, whereas the generic review only presents the general experience of the user toward the item (restaurant, hotel, gadget). Therefore, specific reviews can be effectively used to extract both local and global contexts. In text-based recommendation approaches, the similarity between users is computed using ratings and contexts extracted from their reviews. One of the major issues with such approaches is that they do not consider review characterization for computing user-based similarity. For example, assume that two users write reviews on the same set of items; the first user shares his/her overall experience, whereas the second user provides contextual information by writing on different aspects of the items. In this scenario, both users may be similar in terms of rating because they provided the ratings to a similar set of items. However, they are not similar based on their reviews because they both have expressed completely different aspects of the items. Therefore, we have modeled each review document as a feature vector using a modified set of features given in (Bauman & Tuzhilin, 2014) and described as follows.

- `LogWords`: logarithm of the total words present in a review plus 1.

- `VBDSum`: logarithm of past tense verbs present in a review plus 1.
- `VSum`: logarithm of the total numbers of verbs used in a review plus 1.
- `PronounRatio`: logarithm of the ratio of personal pronouns to the total number of words in a review plus 1.
- `VerbRatio`: ratio of `VBDSum` to `VSum`.

Finally, we have applied the k-means clustering algorithm to categorize review documents into specific and generic categories.

3.2. Local and Global Contextual Information Extraction

In this section, we discuss the local and global contexts extraction process that is later used for generating a collaborative model to learn topic embeddings from specific and generic review documents. We also present a description of different matrices viz. term-document, word co-occurrence, and topic embedding matrix generated using word embedding and topic modeling techniques.

3.2.1. Local Contextual Information Extraction

Local contextual information is embedded within a text corpus and can be extracted using the word embedding approaches, which are neural network-inspired models, such as the CBOW and *skip-gram* model. CBOW predicts the current word using the set of contextual words that appear within a fixed-size window, whereas the skip-gram model predicts the contextual words co-occurring in a window using the current word. For each word w, the skip-gram model learns two embeddings, namely, input embedding (target word embedding) for input matrix $W \in \mathbb{R}^{E \times V}$ and output embedding (context word embedding) for output matrix $W' \in \mathbb{R}^{E \times V}$, where E is the dimension of embedding. The concatenation of these two embedding matrices is equivalent to the word co-occurrence matrix M of size $|V| \times |V|$, as shown in Eq. (1), where each $m_{i,j} \in M$ represents an association between the input and output embeddings.

$$M \approx W^T W' \tag{1}$$

In (Levy & Goldberg, 2014), authors observed that the skip-gram model is optimized when the word co-occurrence matrix M is replaced with the shifted version of the positive point-wise mutual information (SPPMI) matrix, as given in Eq. (2), where w_1 and w_2 represent the current and context words, respectively. The *SPPMI* between w_1 and w_2 is computed using Eq. (3), where point-wise mutual information (PMI) represents the association between w_1 and w_2 in terms of joint and marginal probabilities, as given in Eq. (4).

$$M = SPPMI(w_1, w_2) \tag{2}$$

$$SPPMI_k(w_i, w_j) = max\{PMI(w_i, w_j) \times N - logk, 0\} \tag{3}$$

$$PMI(w_i, w_j) = log\frac{P(w_i, w_j)}{P(w_i)P(w_j)} \tag{4}$$

The objective function for the factorization of SPPMI based on the word co-occurrence matrix M is presented in Eq. (5), where $||W||^2$ and $||W'||^2$ are the regularization terms and λ_t is a parameter used to avoid over-fitting by penalizing the magnitudes of the regularization terms.

$$L_{local} = ||M - W^T W'||^2 + \lambda_t(||W||^2 + ||W'||^2) \tag{5}$$

3.2.2. Global Contextual Information Extraction

Global contextual information extraction represents topics that are embedded in a set of documents and can be extracted using various topic modeling techniques, such as LDA, NMF, and PLSA. In topic modeling, documents are represented as a distribution of topics, and topics are represented as a multinomial distribution of words. In NMF, a document-term matrix $D \in \mathbb{R}^{V \times N}$ is decomposed into two matrices as $D \approx UU'$, where V and N represent the number of terms and documents, respectively; $U \in \mathbb{R}^{K \times N}$ and $U' \in \mathbb{R}^{K \times V}$ represent the document-topic and topic-term matrices, respectively; and K represents the total number of topics. The objective function for the factorization of the document-term matrix D is presented in Eq. (6), wherein the first term represents the minimization of the mean square error, and the second term avoids the over-fitting by penalizing the magnitudes $||U||^2$ and $||U'||^2$.

$$L_{global} = ||D - U^T U'||^2 + \lambda_t(||U||^2 + ||U'||^2) \tag{6}$$

One of the major issues with topic modeling is the instability of the topics, i.e., at each iteration, the topic modeling technique generates different topics from the same set of documents. In this study, we have used NMF-based ensemble topic modeling presented in (Belford et al., 2016) to avoid the instability issue.

3.3. Topic Embedding Learning

In this section, we present the process of topic embedding learning through the collaborative modeling of local and global contextual information discussed in the previous sections. The motive behind including local and global contexts is that the words which are semantically related to each other contribute toward the generation of word embedding and topic modeling. Therefore, in line with (Xun et al., 2017), we learn word embedding and topic modeling from specific and generic reviews. To form a collaborative model and generate topic embedding, the topic-term matrix U' is factorized into context word embedding and topic embedding matrices. The objective function for the factorization of topic-term matrix U' is presented in Eq. (7), where $||T||^2$ and $||W'||^2$ are the topic embedding and context word embedding regularization terms, respectively.

$$L_{cm} = ||U' - T^T W'||^2 + \lambda_t(||T||^2 + ||W'||^2 \tag{7}$$

$$L_{cm} = \sum_{k=1 v=1}^{K,V} (u'_{k,v} - t_k^T w'_v)^2 + \lambda_t(\sum_{k=1}^{K} t_k^T t_k + \sum_{v=1}^{V} w_v'^T w'_v) \tag{8}$$

The collaborative model presented in Eq. (7) shows that the topic-term matrix U' is used to derive the global context model, whereas the context word embedding matrix W' helps to generate the local context and collaborative models. The collaborative model in Eq. (7) can further be generalized using the relevant components of the language model. A generalized form of the collaborative language model is presented in Eq. (8) in which the gradient of the collaborative model can be computed with respect to t_k as follows.

$$\frac{\partial L_{cm}}{\partial t_k} = 2\sum_{v=1}^{V}(u'_{kv} - t_k^T w'_v)(-w'_v) + 2\lambda_t t_k = 0$$

$$\implies \sum_{v=1}^{V}(u'_{kv} - t_k^T w'_v)(-w'_v) + \lambda_t t_k = 0$$

$$\implies t_k = (\sum_{v=1}^{V} w'^T_v w'_v + \lambda_{t_k} I)^{-1}(\sum_{v=1}^{V} u'_{kv} w'_v) \tag{9}$$

Similarly, Eqs. (5) and (6) can further be expanded to compute word and topic embeddings, respectively, which are used later to update t_k in Eq. (9).

3.4. User Similarity Calculation and Rating Prediction

In this section, we present the user similarity calculation and rating prediction using the topic embeddings generated in section 3.3. As discussed earlier, we have used user-based collaborative filtering to identify top-k users that are similar to user u (i.e., U_u^n) for item i. The identified top-k users have their historical data in terms of reviews and ratings on various items. Using Eq. (9), topic embedding vectors $\mathcal{T}_u$ are generated for each user, as given in Eq. (10). Following the generation of the topic embeddings, Eq. (11) calculates the similarity between every pair of users u and v, represented as $TESim(u, v)$, where $Sim()$ is the similarity function which computes *Cosine* similarity using the topic embedding vectors $\mathcal{T}_u$ and $\mathcal{T}_v$; and I_u and I_v represent the sets of items on which users u and v have respectively provided their reviews. Thereafter, in line with (Schafer et al., 2007), the similarity values of the top-k users are combined with their rating values (if available) on the overlapping items of users u and v

using Eq. (12), where U_u^n represents the top-n users that are similar to u, and r_v is the true rating value provided by user v.

$$\mathcal{T}_u = \sum_{j=1,k=1}^{n,K} t_k(rev_{uj}) \tag{10}$$

$$TESim(u,v) = \frac{\sum_{m \in I_u \cap I_v} Sim(\mathcal{T}_{um}, \mathcal{T}_{vm})}{|I_u \cap I_v|} \tag{11}$$

$$\hat{r}_{ui} = \frac{\sum_{v \in U_u^n} TESim(u,v) r_v}{\sum_{v \in U_u^n} |TESim(u,v)|} \tag{12}$$

One major issue with Eq. (12) is that it can not handle biased users who knowingly give high or low ratings to some specific items, affecting the predictions of the recommender systems. Therefore, we have modified Eq. (12) by adding a first-order approximation to nullify the effect of the biased users (Schafer et al., 2007), as shown in Eq. (13), where $b_{ui} = \mu + b_u + b_i$, b_u and b_i are the user and item differences with respect to the overall average ratings of the items, and μ is the items' mean rating.

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in U_u^n} TESim(u,v) r_v}{\sum_{v \in U_u^n} |TESim(u,v)|} \tag{13}$$

Similarly, to avoid overfitting issue, a regularized model presented in (Hu et al., 2008; Koren, 2010) is used to handle the loss function, as shown in Eq. (14), where $\mathcal{K}$ represents the rating set and $||.||$ is the Frobenius norm. The first term in this equation computes the mean square error, which measures the squared difference between the actual and predicted ratings. The regularized model helps to learn the best fit rating values of user deviation b_u and item deviation b_i. From another aspect, the second term constitutes the regularization terms $\lambda_a(||b_u||^2)$ and $\lambda_b(||b_i||^2)$ to avoid the overfitting issue by controlling the size of the parameters.

$$L = min \sum_{(ui) \in \mathcal{K}} (r_{ui} - b_{ui})^2 + \lambda(||b_u||^2 + ||b_i||^2) \tag{14}$$

To compute the values of bias terms b_u and b_i, the authors in (Hu et al., 2008; Koren, 2010) proposed an approach that uses the overall average rating of items. The major issue with the proposed solution is that they are not accurate methods for predicting the ratings. To handle this issue and compute regularization parameters, Stochastic Gradient Descent (SGD) or Alternating Least Squares (ALS) (Zhou et al., 2008) methods can be applied in Eq. (14). In this study, we use ALS to compute the regularization parameters because (i) ALS performs better on sparse data, (ii) it is scalable over large datasets, and (iii) it can perform parallel execution in comparison to other approaches.

4. Experimental Setup and Results

In this section, we explain the experimental evaluations of the proposed `RecTE` approach. We first present a brief explanation of the three datasets used in our experiments. Thereafter, we explain various evaluation metrics, baselines, and state-of-the-art methods, and the performance evaluation of the `RecTE` and other comparative methods. We also present an analysis of top-k users on all three datasets. Finally, we present a comparative evaluation of `RecTE` to handle *cold-start* problem. In summary, we aim to answer the following research questions:

- **RQ1**: How does `RecTE` perform in comparison to the baselines and state-of-the-art methods?
- **RQ2**: What is the impact of finding top-k similar users in terms of rating prediction?
- **RQ3**: How does topic embedding used in the `RecTE` model for rating prediction perform better than word-level and document-level embeddings?

4.1. Dataset Description

To evaluate the effectiveness of `RecTE`, we demonstrate experiments over three real-world datasets – `YelpNYC`, `YelpZip`, and `TripAdvisor`. The first and second datasets are associated with Yelp[1], a crowd-sourced review forum that allows users to write reviews on hotels and restaurants. Yelp contains review information related to restaurants, shopping, nightlife, automotive, home services, beauty and spa, and active life. In this study, we use the restaurant dataset for experimental evaluations. The restaurant review dataset contains information related to user check-ins, business, user information, tip, and user reviews. The third dataset is from TripAdvisor, a Web platform that provides accommodation and booking services for various hotels and restaurants and permits users to share their views. Table 2 shows a brief statistics of the datasets, and a brief description is presented in the following paragraphs.

- YelpNYC: The YelpNYC dataset was collected and used by (Rayana & Akoglu, 2015) and contains reviews on hotels and restaurants in the New York city area. It contains $359,052$ reviews written by $160,225$ users on 923 items. The number of reviews per user and item is 2.24 and 389, respectively.

- YelpZip: The YelpZip dataset used zip code to collect reviews on various hotels and restaurants located in Pennsylvania, New Jersey, Connecticut, and Vermont (Rayana & Akoglu, 2015). It contains $608,598$ reviews written by $260,277$ users on $5,044$ items. The number of reviews per user and item is 2.33 and 120.65, respectively, which is low in comparison to YelpNYC.

- TripAdvisor: The TripAdvisor is a restaurant and hotel-related Web forum which is used for accommodation booking, reviewing, and other travel-related content. We used the TripAdvisor dataset generated in (Wang et al., 2011) for aspect-based rating analysis. It contains $407,416$ reviews written by $290,323$ users on $1,188$ items. The number of reviews per user and item is 1.40 and 342.94, respectively.

[1] https://www.yelp.com/dataset

4.2. Experimental Settings

This section discusses the evaluation metrics, baselines methods, state-of-the-art methods, and parameter settings used to perform experiments and a comparative analysis of `RecTE` with other standard approaches.

4.2.1. Evaluation Metrics

In this study, we have used three different types of standard evaluation metrics, which are briefly described in the following paragraphs.

- *Error-based metrics*: These metrics compute an absolute deviation between the actual and predicted ratings to find errors. We have used two error-based metrics – MAE and RMSE. MAE is calculated as the mean of the prediction errors, which are the deviation between the true values and the predicted values in a test dataset (Herlocker et al., 2004). RMSE, from another aspect, is defined as the standard deviation of the residuals for a test dataset, where residuals represent the deviations between the true values and predicted values. MAE and RMSE are formally defined in Eqs. (15) and (16), respectively. In these equations, $\mathcal{T}$ is the test dataset, $\hat{r}_{xy}$ represents the predicted rating value, and r_{xy} is the true rating value for user x on item y.

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

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

- *Decision support-based metrics*: These metrics determine the ability of recommendation techniques in terms of how well they facilitate their users to make good decisions, wherein 'good decision' aims to recommend relevant items and filter irrelevant items. Precision, Recall, and F-score metrics come under this category, and their values are calculated using the recommended and relevant items. The items which are identified by the recommendation techniques for recommendations are the recommended items, and they contain the predicted rating values. From another aspect, relevant items are those that are consumed by the users in the past, and they contain the true rating values. The relevant items along with their true rating values are used by the recommendation techniques to identify the recommended items. Precision can be defined as the ratio of the total number of recommended relevant items and the recommended items, as given in Eq. (17). Recall is the ratio of the total number of recommended relevant items and total relevant items, as given in Eq. (18). Finally, F-score is the harmonic mean of Precision and Recall, as given in Eq. (19).

$$Precision\ (P) = \frac{\#recommended\ items\ which\ are\ relevant}{\#of\ recommended\ items} \tag{17}$$

$$Recall\ (R) = \frac{\#recommended\ items\ which\ are\ relevant}{\#of\ relevant\ items} \tag{18}$$

$$F - score\ (F) = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{19}$$

- *Ranking-based metrics*: These metrics extend Precision and Recall to take the position of correct items in a ranked list of items. We have used the Normalized Discounted Cumulative Gain (NDCG) metric to evaluate the proposed `RecTE` approach. To compute NDCG, first, we compute the Discounted Cumulative Gain (DCG) to measure the utility of the items at each position in the recommended item list, as defined in Eq. (20). Thereafter, we apply normalization to obtain NDCG, as defined in Eq. (21). In these equations, $m(i)$ is the sorted list of items, $r(j)$ represents the position of item i in $m(i)$, and N_l is the normalized factor of DCG for the sorted list $m(i)$ (Valizadegan et al., 2000). The NDCG value lies

Category	**YelpNYC**	**YelpZip**	**TripAdvisor**
#Users	160,225	260,277	290,323
#Items	923	5,044	1188
#Reviews	359,052	608,598	407,416
Data Sparsity	99.75%	99.95%	99.88%
#Reviews per User	2.24	2.33	1.40
#Reviews per Item	389.00	120.65	342.94

Table 2: Statistics of the datasets

between 0 and 1, where 0 and 1 represent the worst and best ranking of the items, respectively.

$$DCG(i) = \sum_{j \in m(i)} \frac{2^{R_{ij}} - 1}{log_2(1 + r_j)} \tag{20}$$

$$NDCG = \frac{1}{n} \sum_{l=1}^{n} NDCG(l) = \frac{1}{n} \sum_{l=1}^{n} \frac{1}{N_l} DCG(l) \tag{21}$$

4.2.2. Baseline Methods

In this section, we describe baseline methods considered for comparative analysis in this study. To perform experimental evaluations, we have considered 9 baselines viz. Co-clustering, Slope One, two variants of the k-Nearest Neighbors (kNN), Singular-Value Decomposition (SVD), SVD++, Normal Predictor, Baseline, and Non-negative Matrix Factorization (NMF) that are briefly described in the following paragraphs.

- Co-clustering: It considers the pairwise association of two coincident entities. Eq. (22) presents the formulation of co-clustering technique for rating estimation, where $\overline{u}$ and $\overline{j}$ are the users' and items' mean value, respectively, $\overline{cl_{uj}}$ is the overall average rating of coincident clusters, and $\overline{cl_u)}$, $\overline{cl_j)}$ are the average ratings of the users' and items' clusters (George

& Merugu, 2005).

$$\hat{r}_{uj} = \overline{cl_{uj}} + (\overline{u} - \overline{cl_u}) + (\bar{j} - \overline{cl_j}) \tag{22}$$

- Slope One: This recommendation technique is used for predicting the rating values using the item-based collaborative filtering approach. The mean ratings of the users and items are used for rating estimation. Eq. (23) presents the formulation of the Slope One method, where R_m represents an apposite item set and $dev_{m,n}$ is the rating deviation between items m and n.

$$\hat{r}_{vm} = \mu_v + \frac{1}{cardinality(R_m)} \sum_{n \in R_m} dev_{m,n} \tag{23}$$

- KNN: As the name suggests, this method uses top-k similar users or items for rating estimation. The recommendation techniques that use KNN, first identify the k similar users or items using a similarity function. Thereafter, the true rating values of the identified users or items are used for rating estimation. Eq. (24) presents the formulation of KNN method, where r_{aj} represents the rating of user a on item j, and k is top-k users or items. In centered-KNN, the mean values of users or items are added to the predicted ratings.

$$\hat{r}_{aj} = \frac{\sum_{l \in M_a^k j} Sim(j,l).r_{al}}{\sum_{l \in M_a^k j} Sim(j,l)} \tag{24}$$

- SVD++, SVD, and NMF: These methods use the concept of matrix factorization. They are modeling-based recommendation techniques used in collaborative filtering. In these methods, the user-item interaction matrix that contains the true rating values is split into two low-dimensional matrices viz. *user interest* and *item features* matrices. The split matrices are further used to estimate the ratings of the unrated items.

- Normal Predictor: It uses the idea of maximum likelihood estimation for rating prediction. Eqs. (25) and (26) present the formulation of a normal predictor, where σ and μ are the variance and mean values, respectively, which are used to calculate the normal distribution, R_{train} is the training dataset, and r_{mn} represents the rating of user m on item n.

$$\mu = \frac{1}{|R_{train}|} \sum_{r_{mn} \in R_{train}} r_{mn} \tag{25}$$

$$\sigma = \sum_{r_{mn} \in R_{train}} \frac{(r_{mn} - \mu)}{|R_{train}|} \tag{26}$$

- *Baseline*: It uses the users' and items' rating deviations with respect to the overall mean ratings of the items in a dataset, as given in Eq. (27), where dev_j and dev_u are the deviations of item j and user u, and μ is the overall mean rating of the items.

$$\hat{r}_{uj} = \mu + dev_j + dev_u \tag{27}$$

4.2.3. State-of-the-Art Methods

This section presents a detailed description of the five state-of-the-art methods viz. `DARMH`, `Recop`, `CUNE-MF`, `AutoRec`, and `AESR`, which are evaluated and compared with our proposed `RecTE` approach over all three benchmark datasets.

- `DARMH` (Liu et al., 2021): It is a deep learning-based multi-task recommender system which uses reviews and review helpfulness votes to predict ratings and review helpfulness. DARMH introduced a dual attention mechanism comprising of local and interactive attentions. The local attention is used to extract key features from review embeddings, whereas the interactive attention is used to extract the personalized preferences of a particular user for a particular item.

	YelpNYC		YelpZip		TripAdvisor	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
RecTE	0.7136	0.8659	0.7835	0.9995	0.7512	0.9453
DARMH (Liu et al., 2021)	0.7305	0.8794	0.8179	1.0439	0.775	0.9827
Recop (Bathla et al., 2021)	0.7929	1.0472	0.8865	1.1124	0.8208	1.0552
CUNE-MF (Zhang et al., 2017)	0.7438	0.9081	0.8211	1.0486	0.7815	0.9870
AutoRec (Sedhain et al., 2015)	0.7384	0.8894	0.8158	1.0315	0.7752	0.9812
AESR (Nisha & Mohan, 2018)	0.7312	0.8807	0.8492	1.0719	0.7860	0.9906
KNN	0.8350	1.1261	0.9237	1.2182	0.8119	1.0587
NMF	0.8909	1.1641	0.9734	1.2492	0.8411	1.0652
SVD	0.7911	1.036	0.8884	1.1060	0.8117	1.0361
SVD++	0.7905	1.0223	0.8734	1.0924	0.8067	1.0268
Co-clustering	0.8325	1.1208	0.9367	1.2221	0.8114	1.0385
Slope One	0.8382	1.1303	0.9479	1.2348	0.8388	1.0582
Centered-KNN	0.8315	1.1186	0.9172	1.2115	0.8099	1.0549
Baseline	0.7708	1.0216	0.89364	1.1136	0.8125	1.0382
Normal Predictor	1.071	1.3817	1.1939	1.5255	0.8749	1.1516

Table 3: Comparative evaluation results of `RecTE`, baselines, and state-of-the-art methods in terms of MAE and RMSE values over `YelpNYC`, `YelpZip`, and `TripAdvisor` datasets

- `Recop` (Bathla et al., 2021): It is a user-based collaborative filtering recommendation model where users' ratings and reviews are used for rating prediction. Recop uses users' sentiments and numeric ratings to compute the similarity between the users, and thereafter it uses k-nearest neighbors based on the similarity scores for rating prediction.

- `CUNE-MF` (Zhang et al., 2017): It is a collaborative filtering-based matrix factorization model which uses users' social information to generate a user-based collaborative network. The generated collaborative network is used to extract implicit social information of the users using the embedding nodes constructed using rating data. A user-based similarity matrix is generated using the implicit social information, and used to predict the ratings of the unrated items.

- `AutoRec` (Sedhain et al., 2015): It is an autoencoder-based framework which uses collaborative filtering network to predict the rating of items by minimizing the autoencoder error, as given in Eq. (28). In this equation, $r^{(j)}$ is the item rating, $t(r^{(j)};\theta)$ is the reconstruction rating of item j using autoencoder, and M and N are the weights of the encoder and decoder layers, respectively.

$$min\sum_{j=1}^{n}||r^{(j)} - t(r^{(j)};\theta)||^2 + \frac{\lambda}{2}(||M||^2 + ||N||^2) \tag{28}$$

- `AESR` (Nisha & Mohan, 2018): It is a deep learning-based recommender system, where deep autoencoder and semantic social information of users are integrated to optimize the rating prediction function. Eq. (29) presents the joint optimization function for AESR, where r_{kl} is the rating given by user k on item l, $t(r_{kl};\theta)$ is the reconstruction rating of item l, $\lambda(\theta)$ is the regularizing term, and $f(p_k;\phi_p$ and $f(p_s;\phi_2$ represent the user and item encoder values, respectively.

	YelpNYC				YelpZip				TripAdvisor			
	Precision	Recall	F-score	NDCG	Precision	Recall	F-score	NDCG	Precision	Recall	F-score	NDCG
RecTE	0.9223	0.8078	0.8612	0.8605	0.7814	0.7092	0.7435	0.6172	0.8427	0.7362	0.7858	0.7861
DARMH (Liu et al., 2021)	0.9028	0.7922	0.8436	0.8305	0.7492	0.6811	0.7136	0.5863	0.8343	0.7211	0.7736	0.7680
Recop (Bathla et al., 2021)	0.8422	0.7291	0.7822	0.7452	0.6682	0.5819	0.6221	0.4813	0.7821	0.6710	0.7238	0.6792
CUNE-MF (Zhang et al., 2017)	0.8823	0.7672	0.8207	0.7961	0.7402	0.6845	0.7112	0.5772	0.8051	0.7018	0.7498	0.7571
AutoRec (Sedhain et al., 2015)	0.8995	0.7758	0.8330	0.8117	0.7521	0.6890	0.7191	0.5854	0.8164	0.7119	0.7605	0.7627
AESR (Nisha & Mohan, 2018)	0.9015	0.7887	0.8413	0.8269	0.7218	0.6508	0.6843	0.5619	0.7991	0.6918	0.7415	0.7480
KNN	0.8315	0.7311	0.7753	0.7668	0.6314	0.5925	0.6113	0.4884	0.7664	0.6619	0.7102	0.6714
NMF	0.8251	0.7336	0.7766	0.7541	0.6518	0.5560	0.6001	0.4854	0.7712	0.6497	0.7051	0.6711
SVD	0.8569	0.7381	0.7930	0.7727	0.6819	0.6024	0.6396	0.5115	0.7954	0.6745	0.7299	0.6925
SVD++	0.8583	0.7390	0.7938	0.7843	0.6917	0.6047	0.6452	0.5206	0.7991	0.6794	0.7343	0.7012
Co-clustering	0.8109	0.7292	0.7678	0.7644	0.6527	0.5941	0.6219	0.4870	0.7685	0.6658	0.7134	0.6792
Slope One	0.8023	0.7112	0.7539	0.7460	0.6458	0.5830	0.6127	0.4910	0.7412	0.6382	0.6858	0.6847
Centered-KNN	0.8358	0.7381	0.7839	0.7701	0.6471	0.6005	0.6222	0.5043	0.7705	0.6678	0.7154	0.6781
Baseline	0.8663	0.7470	0.8022	0.7865	0.6264	0.5912	06082	0.5094	0.7682	0.6598	0.7098	0.6851
Normal Predictor	0.7809	0.6719	0.7226	0.7518	0.5604	0.4943	0.5253	0.4478	0.7029	0.5968	0.6454	0.6245

Table 4: Comparative evaluation results of `RecTE`, baselines, and state-of-the-art methods in terms of Precision, Recall, F-score, and NDCG values over `YelpNYC`, `YelpZip`, and `TripAdvisor` datasets

$$L = \frac{1}{2}\sum_{k=1}^{r}\sum_{l=1}^{r}||r_{(kl)} - t(r_{(kl)};\theta)||^2 + \lambda(\theta) + \frac{\gamma}{2}\sum_{k=1}^{m}\sum_{s\in S(k)}||f(p_k;\phi_p) - f(p_s;\phi_2)||_F^2 \quad (29)$$

4.2.4. Parameter Settings

In this section, we discuss the role and settings of various parameters, such as *learning rate*, *batch size*, *window size*, *number of iterations*, and *embeddings.* We have empirically tested the values of the parameters and reported the best values for which the predicted values of the unrated items are minimum. In this study, we set the size of context window to 10 which considers 5 preceding and 5 following words in the context window of a target word. The embedding size for topic (T) and word (W) embeddings is set to 50, whereas the number of topics (K) in each embedding is set to 20. The values of the λ_d and λ_w parameters that control the weights, and λ_w parameter which is used for regularization are set as $\lambda_d = 1e - 1$, $\lambda_w = 2e - 2$, and $\lambda_w = 1e - 5$.

4.3. Performance Evaluation and Comparison Results(RQ1)

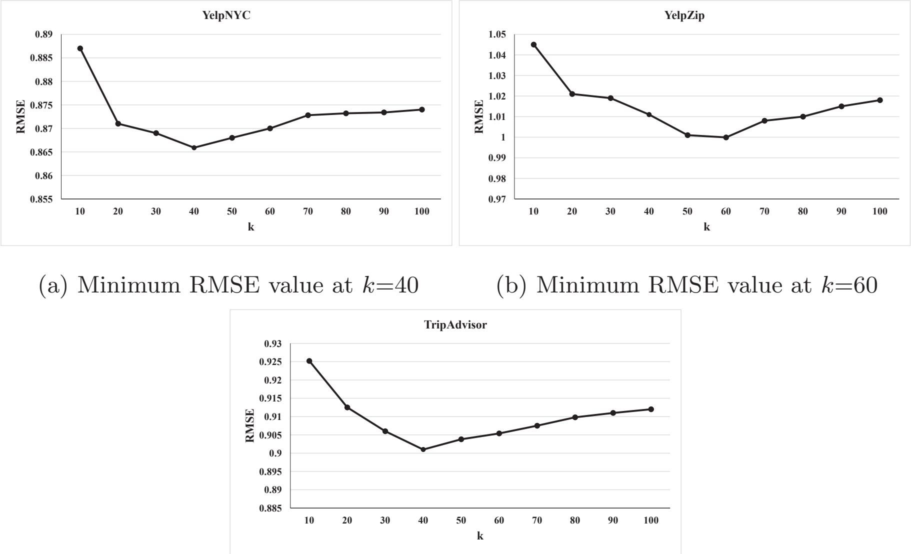
In this section, we discuss the comparative evaluation results of RecTE against all baselines and state-of-the-art methods. In order to implement baselines and state-of-the-art algorithms, we have used two python libraries viz. SurPRISE (Simple Python Recommendation System Engine) (Hug, 2017) and RecQ (Yu et al., 2018). RecQ implements both rating prediction and ranking-based recommender systems. Table 3 presents the performance evaluation and comparison results in terms of MAE and RMSE values. In this table, YelpZIP dataset shows high MAE and RMSE values because the number of reviews per item in this dataset is minimum in comparison to the YelpNYC and TripAdvisor datasets. It can also be noted that SVD++ outperformed all other baselines which is mainly due to the fact that SVD++ considers implicit ratings (implicit feedback information) in rating prediction. It can also be observed from this table that RecTE outperforms SVD++ method in terms of both MAE and RMSE values by 9.72% and 15.29% on YelpNYC dataset, 10.29% and 8.50% on YelpZIP dataset, and 6.87% and 7.93% on TripAdvisor dataset. Similarly, table 4 presents the performance evaluation and comparison results in terms of the ranking-based and decision support-based metrics. It can be observed from this table that RecTE beats SVD++ with an improved Precision, Recall, F-score, and NDCG values by 6.93%, 8.51%, 7.82%, and 8.85% on YelpNYC dataset, 11.47%, 14.73%, 13.22%, and 15.65% on YelpZIP dataset, and 5.17%, 7.71%, 6.55%, and 10.8 on TripAdvisor dataset.

As discussed earlier, RecTE is compared with five state-of-the-art algorithms – DARMH (Liu et al., 2021), Recop (Bathla et al., 2021), CUNE-MF (Zhang et al., 2017), AutoRec (Sedhain et al., 2015), and AESR (Nisha & Mohan, 2018), and the comparison results are shown in the same tables, i.e., tables 3 and 4.DARMH and Recop use both numerical ratings and textual information in there proposed algorithms. It may be noted that the user-based collaborative network in the rest of state-of-the-art methods viz. CUNE-MF, Autorec, and AESR only uses user ratings for recommendation, whereas RecTE uses both user ratings and textual information for generating the collaborative network.

It can be observed from these tables that `DARMH`, `Recop`, `CUNE-MF`, `Autorec`, and `AESR` only perform better on `YelpNYC` and `TripAdvisor` datasets, but not on the `YelpZIP` dataset. This is because `YelpZip` has high data sparsity and suffers with item-based cold-start problem. On analysing the results shown in tables 3 and 4, it can be observed that `RecTE` beats `DARMH`, `Recop`, `CUNE-MF`, `Autorec`, and `AESR` methods on all three datasets. `DARMH` outperforms all other state-of-the-art approaches because it uses both reviews and ratings. The inclusion of textual information helps to find most similar user and items. On the other hand, the rest of state-of-the-art methods performs better in comparison to `Recop`. This is mainly due to the fact that `Recop` does not consider the similarity score between users for rating prediction and only uses the average ratings of similar user. The `RecTE` on average improves MAE and RMSE values by 2.36% and 1.55% on `YelpNYC` dataset, 4.39% and 4.44% on `YelpZIP` dataset, and 3.16% and 3.95% on `TripAdvisor` dataset, in comparison to the state-of-the-art algorithms. Similarly, `RecTE` also outperforms all state-of-the-art algorithms in terms of the decision support-based and ranking-based evaluation metrics.

4.4. Effects of the 'k' Parameter (RQ2)

The parameter k is an important factor for the rating prediction of `RecTE`, because it determines the number of top-k similar users for each user. The question related to the parameter k is that what should be its optimal value to get minimum MAE and RMSE values? The small value of k results in few users that are not sufficient to provide enough information for a target user. From another aspect, a large value of k may introduce users who are not quite relevant to the target user, and thereby adding noise to the recommendation model. Therefore, there is a strong chance that the performance of the recommendation model degrades. We empirically investigate the best value of k for the proposed `RecTE` model on all three datasets to get minimum RMSE value. The empirical analysis result is presented in figure 3. It can be observed from this figure that with increasing value of k, RMSE decreases first because of the availability of similar users for a target user. But as the value of k increases beyond a



(a) Minimum RMSE value at k=40

(b) Minimum RMSE value at k=60

(c) Minimum RMSE value at k=40

Figure 3: Impact of k to achieve minimum RMSE over YelpNYC, YelpZip, and TripAdvisor datasets

threshold, RMSE starts increasing because of the inclusion of the users who are not relevant (similar) to the target user. It can be observed from figures 3(a), 3(b), and 3(c) that the minimum RMSE values for the datasets YelpNYC, YelpZIP, and TripAdvisor are at k= 40, k = 60, and k = 40, respectively. The reason for high k for YelpZip dataset is that most of the items in the dataset are suffering with cold-start issue because the number of ratings and reviews per item is low. Therefore, a high number of similar users are required to form a collaborative network which is helpful to predict ratings for the target user.

4.5. Effects of Different Embedding Techniques (RQ3)

In this section, we compare the effectiveness of the topic-based, word-based, and document-based embeddings for recommendations and rating predictions that are briefly described in the following paragraphs.

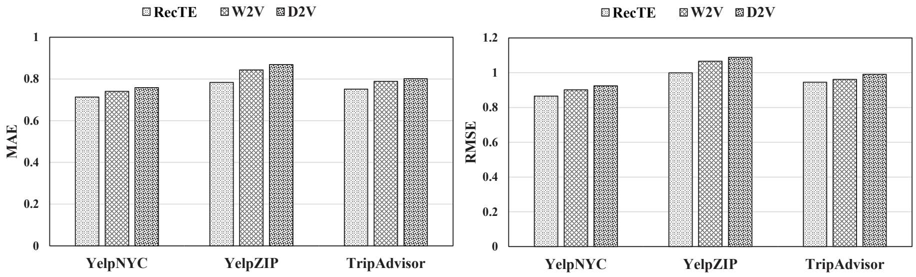
- Word2Vec (W2V): It is a word-based embedding technique which captures the semantic and syntactic similarity and context of a word with other words in a document. The embeddings using W2V can be generated using CBOW model or *skip-gram* model. The CBOW predicts words using its contextual words, whereas *skip-gram* model predicts the contextual words for a given word.

- Doc2Vec (D2V): It is a document-based embedding technique which computes the feature vector of every document in a corpus. D2V is an extension of W2V, and it can also be obtained using either CBOW or *skip-gram* model.

We developed and evaluated recommender systems for both W2V and D2V models over all three datasets using the Gensim library. The MAE and RMSE values of the W2V and D2V-based recommender systems are shown in figure 4. Both word-based and document-based embeddings are learned using the review documents written by the users. It may be noted that the embeddings generated by W2V and D2V depend on the quality of the reviews. It can be observed from figure 4 that RecTE performs better in comparison to W2V-based and D2V-based recommender systems. On analysis, we found that the reason behind the improved performance of RecTE is that it learns topic embeddings on both specific and generic reviews. As a result, RecTE categorizes users based on their reviews, and accordingly it treats users having reviews enriched with contextual features separately from other users. Moreover, RecTE considers words that are both frequently co-occurring and contributes globally in the document, resulting in an improved topic embeddings in comparison to the W2V-based and D2V-based embeddings.

4.6. Dealing with Cold-Start Users

The *cold-start* problem is related to both the availability of ratings received on items from various users and the ratings given by users on various items. The rating prediction for such users and items is a challenging task. In recommender systems, profile generation for *cold-start* users is a difficult task due to



(a) MAE

(b) RMSE

Figure 4: Performance comparison results of `RecTE`, `Word2Vec` (`W2V`), and `Doc2Vec` (`D2V`) at $k = 60$ in terms of MAE and RMSE values over `YelpNYC`, `YelpZip`, and `TripAdvisor` datasets

the minimal availability of rating information for such users. Similarly, descriptive data for *cold-start* items are also limited because the interaction of such items with users is minimum. The lack of availability of data hampers the performance of recommender systems. In this study, since we are using user-based collaborative filtering, we have considered only *cold-start* users for evaluating the performance of `RecTE` towards rating prediction.

4.6.1. `RecTE` *vs. Baselines and State-of-the-art Methods*

Since `RecTE` uses user-based collaborative filtering for rating prediction, we have identified those users who have rated maximum five items and considered them as *cold-start* users (Massa & Avesani, 2007; Jamali & Ester, 2009) to evaluate the performance of `RecTE` and other recommendation techniques. The comparative evaluation results are presented in figures 5, 6, and 7. Since the data sparsity for *cold-start* users is very high, both MAE and RMSE values for such users are high. It can be observed from these figures that SVD++ beats all baselines on all three datasets because it considers implicit ratings. Similarly, out of the five state-of-the-art recommendation techniques, `DARMH` beats other four techniques on `YelpNYC`, `YelpZip`, and `TripAdvisor` datasets. However, `RecTE` beats both SVD++ and state-of-the-art algorithms because the user-based collaborative network in `RecTE` uses both local and global contextual information extracted from the reviews. Unlike `DARMH`, this helps to pay attention on those features which are locally and globally important. Further, the users who either provide ratings or reviews are able to be a part of the collaborative network. RecTE beats SVD++ in terms of MAE and RMSE values and showed an improvement of 8.74% and 15.43% on YelpNYC dataset, 11.99% and 10.24% on YelpZip dataset, and 5.75% and 5.62% on TripAdvisor dataset. Similarly, in comparison to DARMH, RecTE showed improvement of 1.18% and 1.51% in MAE and RMSE values on YelpNYC dataset. Further, RecTE improved MAE and RMSE values by 2.01% and 2.32%, and 1.68% and 1.09% in comparison to DARMH on YelpZip and TripAdvisor datasets, respectively.

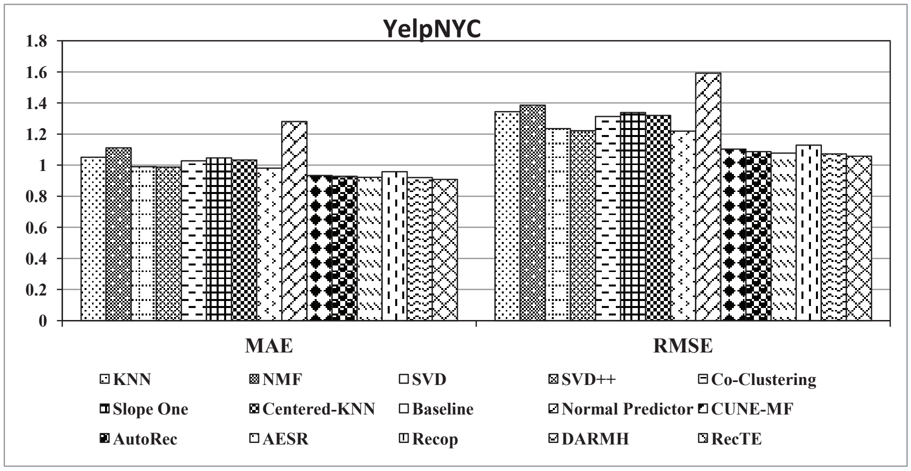


Figure 5: Comparative evaluation results of RecTE, baselines, and state-of-the-art methods in terms of MAE and RMSE values for the *cold-start* users over YelpNYC dataset

5. Conclusion and Future Work

In this study, we have proposed the development of a topic embedding-based recommender system, RecTE. This system learns topic embeddings with the help of word embeddings and topic modeling, and uses them along with rating data for predicting the ratings of the unrated items. The novelty of RecTE lies in predicting ratings using topic embedding learned by incorporating local and global contextual information extracted from the reviews and integrating them with

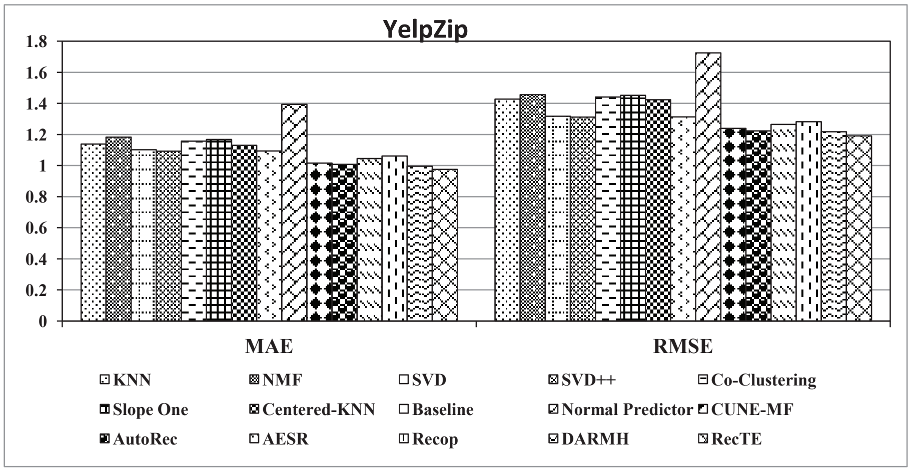


Figure 6: Comparative evaluation results of `RecTE`, baselines, and state-of-the-art methods in terms of MAE and RMSE values for the *cold-start* users over `YelpZip` dataset

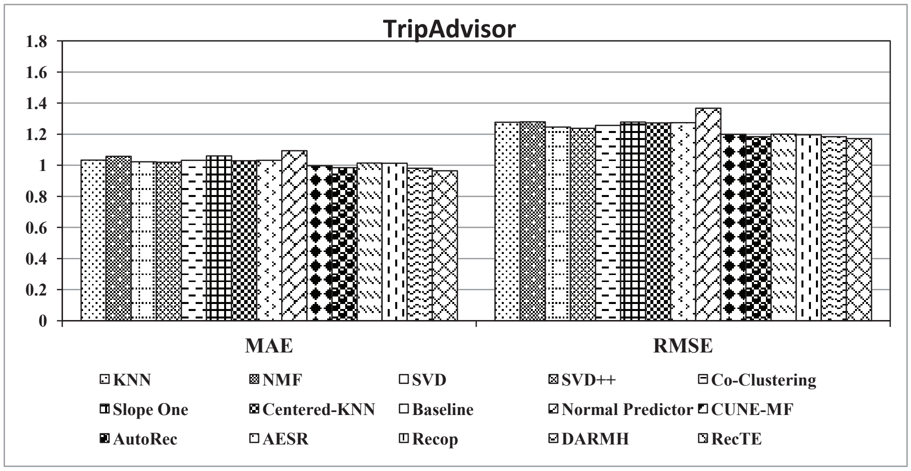


Figure 7: Comparative evaluation results of `RecTE`, baselines, and state-of-the-art methods in terms of MAE and RMSE values for the *cold-start* users over `TripAdvisor` dataset

the user-based collaborative filtering. Based on the POS tag-based features, reviews are first classified into specific and generic classes using k-means clustering. Specific reviews contain user experience on different aspects of the items, whereas generic reviews show the overall experience of a user about the item itself. Thereafter, a collaborative model is used to learn topic embeddings by incorporating local and global contextual information identified through word embedding and ensemble-based topic modeling techniques. The topic embeddings are used to find top-k similar users for each target user for rating prediction. The `RecTE` is validated over three real-world datasets and compared with nine baselines and five state-of-the-art approaches using different evaluation metrics. The empirical evaluation results reveal that `RecTE` significantly improves the rating prediction accuracy and beats the baselines and state-of-the-art recommendation techniques. `RecTE` also improves the rating prediction accuracy for the *cold-start* users in comparison to other standard recommendation approaches. In summary, this study provides new insight by exploiting users' reviews and ratings to compute their similarities and thereby improve the accuracy of the recommender system. Enhancing `RecTE` by incorporating other deep learning techniques, such as `LSTM` and `BiLSTM`, to model user reviews and classifying them using deep learning-based classifiers seem promising directions for future research.

References

Bathla, G., Singh, P., Kumar, S., Verma, M., Garg, D., & Kotecha, K. (2021). Recop: fine-grained opinions and sentiments-based recommender system for industry 5.0. *Soft Computing*, (pp. 1–10).

Bauman, K., & Tuzhilin, A. (2014). Discovering contextual information from user reviews for recommendation purposes. In *Proceddings of the Workshop on New Trends in Content based Recommender Systems (CBRecSys'14)* (pp. 2–9). Silicon Valley, USA: ACM.

Belford, M., MacNamee, B., & Greene, D. (2016). Ensemble topic modeling via matrix factorization. In *Proceedings of the 24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16)*. Dublin, Ireland: CEUR Workshop Proceedings volume 1751.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*, 993–1022.

Chen, C., Zheng, X., Wang, Y., Hong, F., & Lin, Z. (2014). Context-aware collaborative topic regression with social matrix factorization for recommender systems. In *Proceddings of the 28th AAAI Conference on Artificial Intelligence* (pp. 1–7). Quebec, Canada: AAAI.

Cheng, Z., Shen, J., Kankanhalli, M., & Nie, L. (2017). Exploiting music play sequence for music recommendation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)* (pp. 3654–3660). Melbourne, Australia: ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

George, T., & Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the 5th International Conference on Data Mining (ICDM'05)* (pp. 625–628). Houston, USA: IEEE.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*, 5–53.

Hu, B., & Ester, M. (2013). Spatial topic modeling in online social media for location recommendation. In *Proceddings of the 7th International Conference on Recommender Systems (RecSys'07)* (pp. 25–32). Hong Kong, China: ACM.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *In Proceedings of 8th International Conference on Data Mining (ICDM'08)* (pp. 263–272). Pisa, Italy: IEEE.

Huang, E. H., Socher, R., Manning, C. D., & Ng, A. (2008). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)* (pp. 873–882). Jeju Island, Korea: ACM.

Hug, N. (2017). Surprise, a Python Library for Recommender Systems http://surpriselib.com.

Jamali, M., & Ester, M. (2009). Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining (KDD'09)* (pp. 397–406). Paris, France: ACM.

Jing, L., Wang, P., & Yang, L. (2015). Sparse probabilistic matrix factorization by llaplace distribution for collaborative filtering. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)* (pp. 1771–1777). Buenos Aires, Argentina: ACM.

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *4*, 1.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)* (pp. 2177–2185). Montreal, Canada: ACM.

Liu, D., Li, J., Du, B., Chang, J., & Gao, R. (2019). Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceddings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19)* (pp. 344–352). Anchorage, USA: ACM.

Liu, L., Tang, L., Dong, W., Yao, S., & Zhou, W. (2016). An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, *5*, 1937–1960.

Liu, Z., Yuan, B., & Ma, Y. (2021). A multi-task dual attention deep recommendation model using ratings and review helpfulness. *Applied Intelligence*, (pp. 1–13).

Manotumruksa, J., Macdonald, C., & Ounis, I. (2016). Regularising factorised models for venue recommendation using friends and their comments. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM'16)* (pp. 1981–1984). Indianapolis, USA: ACM.

Massa, P., & Avesani, P. (2007). Trust metrics on controversial users: Balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems*, *3*, 39–64.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceddings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)* (pp. 3111–3119). Lake Tahoe, Nevada: ACM.

Musto, C., Franza, T., Semeraro, G., de Gemmis, M., & Lops, P. (2018). Deep content-based recommender systems exploiting recurrent neural networks and linked open data. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization* (pp. 239–244). ACM.

Musto, C., Semeraro, G., de Gemmis, M., & Lops, P. (2016). Learning word embeddings from wikipedia for content-based recommender systems. In *European Conference on Information Retrieval* (pp. 729–734). Springer.

Nisha, C. C., & Mohan, A. (2018). A social recommender system using deep architecture and network embedding. *Applied Intelligence*, *49*, 1937–1953.

Ozsoy, M. G. (2016). From word embeddings to item recommendation. *arXiv preprint arXiv:1601.01356*, .

Pena, F. J., Morgan, D. O., Tragos, E. Z., Hurley, N., Duriakova, E., Smyth, B., & Lawlor, A. (2020). Combining rating and review data by initializing latent factor models with topic models for top-n recommendation. In *Proceedings of the 14th ACM Conference on Recommender System (RecSys' 20)* (pp. 438–443). Brazil: ACM.

Rayana, S., & Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)* (pp. 985–994). Sydney, Australia: ACM.

Schafer, J. B., Frankowski, D., Herlocker, J. L., & Sen, S. (2007). Collaborative filtering recommender systems. *The Adaptive Web*, *4321*, 291–324.

Sedhain, S., Menon, A. K., Sanner, S. P., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In *Proceddings of the 24th International Conference on World Wide Web (WWW'15)* (pp. 111–112). Florenance, Italy: ACM.

Sejwal, V. K., & Abulaish, M. (2021). Camo: A context-aware movie ontology generated from lod and movie databases. *Multimedia Tools and Applications*, *80*, 7247–7269.

Sejwal, V. K., Abulaish, M., & Jahiruddin (2020). Crecsys: A context-based recommender system using collaborative filtering and lod. *IEEE Access*, *8*, 158432–158448.

Sheu, H.-S., & Li, S. (2020). Context-aware graph embedding for session-based news recommendation. In *Proceedings of the 14th ACM Conference on Recommender System (RecSys'20)* (pp. 657–662). Brazil: ACM.

Tan, Y., Zhang, M., Liu, Y., & Ma, S. (2016). Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)* (pp. 2640–2646). New York, USA: ACM.

Valizadegan, H., Jin, R., Zhang, R., & Mao, J. (2000). Learning to rank by optimizing ndcg measure. *SIGIR*, (pp. 41–48).

Wang, C., & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In *Proceddings of the 17th International Conference on Knowledge Discovery and Data Mining (KDD'11)* (pp. 448–456). California, USA: ACM.

Wang, H., Lu, Y., & Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'11)* (pp. 618–626). ACM.

Xun, G., Li, Y., Gao, J., & Zhang, A. (2017). Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)* (pp. 535–543). Halifax, NS, Canada: ACM.

Yu, J., Gao, M., Li, J., Yin, H., & Liu, H. (2018). Adaptive implicit friends identification over heterogeneous network for social recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 357–366). ACM.

Zhang, C., Yu, L., Wang, Y., Shah, C., & Zhang, X. (2017). Collaborative user network embedding for social recommender systems. In *Proceedings of* *the 2017 SIAM International Conference on Data Mining (SDM'17)* (pp. 381–389). Houston, USA: SIAM.

Zhang, L., Liu, P., & Gulla, J. A. (2019). Dynamic attention-integrated neural network for session-based news recommendation. *Machine Learning*, *108*, 1851–1875.

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *In Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management* (pp. 337–348). Springer.