Int. j. inf. tecnol. (December 2023) 15(8):4243–4257
https://doi.org/10.1007/s41870-023-01458-6

ORIGINAL RESEARCH

SSC-CF: Semantic similarity and clustering-based collaborative filtering for expert recommendation in community question answering websites

Aarthi Paramasivam[1] · S. Jaya Nirmala[1]

Received: 6 March 2023 / Accepted: 26 August 2023 / Published online: 28 September 2023

Abstract Community question answering forums allow users to find knowledge on a topic of interest by asking questions and getting answers from experts. However, it can be challenging to find experts who are knowledgeable in a particular subject, especially when there are millions of questions and thousands of new queries every day.This paper proposes a novel expert recommendation system called Semantic Similarity and Clustering-based Collaborative Filtering (SSC-CF). SSC-CF addresses two key drawbacks of collaborative filtering: scalability and sparsity. Sparsity is addressed by using matrix factorization. In matrix factorization, latent features are identified to detect similarity and generate a prediction based on both the question and the user entities. Whereas a clustering method is employed to group users and questions with shared interests to address scalability. The recommendation system's accuracy is further improved by incorporating semantic similarity. SSC-CF is evaluated on three Stack Exchange sites: gaming, physics, and scifi. The results clearly show that the proposed technique, SSC-CF, is effective in addressing both scalability and sparsity.

Keywords Community question answering · Expert recommendation · Natural language processing · Sparsity · Scalability

1 Introduction

The growth of Web 2.0 has contributed to the rise in popularity of user-generated content-based platforms. Community Question-Answering (CQA) websites like Stack Overflow,[1] Quora,[2] and Yahoo! Answers[3] have grown in popularity in recent years [1]. People now frequently look for information on CQA. The community's members are allowed to respond to queries posed by users and contribute their own. A CQA question usually consists of three parts. First, the question's subject briefly describes the question. The subject helps experts scan the questions and find those that are of interest to them. Then comes a section that describes the specifics of a question. The body part typically contains information about the subject and serves as a supplement to the subject. Finally, when a question is proposed, the question askers assign the corresponding tags. The person who asks the question need to wait for the experts to respond. When multiple responses are received to a question, the asker can select the best response as the accepted answer. While CQA services are a great resource for anyone looking for information, their fast expansion presents some special difficulties [2]. These websites receive thousands of new questions daily in addition to the millions of questions that currently exist. Finding a question that matches a respondent's area

✉ Aarthi Paramasivam
paramasivamaarthi@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli 620015, Tamilnadu, India

[1] stackoverflow.com.

[2] www.quora.com.

[3] answers.yahoo.com.

of expertise is, therefore, challenging. After posting their inquiries, users anticipate high-quality responses. Hence several recommendation approaches are used to get more relevant individuals to contribute their thoughts and solutions. As a result, recent research on CQA systems has preferred to use recommendation algorithms and certainly associated data mining approaches to bring more active and intelligent solutions to help identify experts automatically. With such suggestions, there is a greater likelihood that each query will be promptly assessed by the appropriate users and will receive more insightful responses.

By recommending personalised and pertinent material to users across a variety of sectors, including movies [3, 4], books [5], and online commerce [6], recommendation systems play a crucial role in improving the user experience. The classification of Recommender Systems (RS) includes three subcategories: Collaborative Filtering (CF) RS [7, 8], content-based RS, and hybrid RS [9]. The user profile and the similarity of the question description are taken into account when the content-based technique makes suggestions. Based on the views of other users, CF provides suggestions to users. Because it won't be concerned with the question's description when making recommendations, it may offer sophisticated recommendations. Because of this property, CF has become a popular filtering approach and is crucial in many applications. Combining CF with content-based systems is what is known as a hybrid system. The primary problem with the CF recommender method is data sparsity [10], which is compounded by the cold start issue [11]. In the absence of complete information, the CF model struggles to make effective recommendations. A sparsity problem occurs when a user interacts with a small percentage of the questions in a specific application domain. Another issue in collaborative filtering is the cold start problem. This is because there is a scarcity of information about new entities, such as new users. When a new user was added to the system, she or he had no voting score, and the system could not determine the relevance to the question.

In this study, a novel approach called SSC-CF is proposed in order to address the sparsity and cold-start problems in CF algorithms and thereby improve the performance of expert recommender systems. This study makes an effort to develop a new CF-based recommendation system using dimensionality reduction and semantic similarity techniques. The suggested system solves the cold-start issue of recommender systems by utilizing Singular Value Decomposition++(SVD++) [12] and semantic similarity to increase prediction accuracy. The proposed system, SSC-CF, is built on the user- and question-based CF. For the question- and user-based CF recommendation portion, SSC-CF utilizes SVD++, and for the question-based semantic similarity calculation, it utilizes Bidirectional Encoder Representations from Transformers(BERT) embeddings with cosine similarity. The contributions of this work can be summarized as follows:

- In this work, we propose a novel expert recommendation system, SSC-CF, that relies on dimensionality reduction and semantic similarity methods.
- The accuracy of expert recommendation systems will increase as a result of solving the sparsity problem in CF.
- The cold-start problem of expert recommendation systems can be solved with dimension reduction strategies.

The article is structured as follows: The recent research on CQA recommendation algorithms is introduced in Sect. 2. The statement of the problem for the proposed work SSC-CF is provided in Sect. 3. Section 4 talks about the suggested SSC-CF approach. The experimental findings are presented in Sect. 5, and the final section summarises the suggested work and potential future directions.

2 Related work

In the information retrieval community, question routing has received a lot of interest recently [13]. This section covers earlier studies on question routing and categorizes the methods into different groups: classification method, language models, topic model, network-based method, and collaborative filtering.

Classification Method: When considering experts as a specific class of users among all users, the issue of recognizing the experts can quickly be changed into a classification issue that seeks to separate such a specific class of expert users from the other users. Comparatively speaking to the other methods, categorization methods are more flexible in how they can apply a variety of features to the expert recommendation problem from the user's perspective, including questions, answers, and user-user interactions. The most popular classification approach for separating experts from non-experts is a Support Vector Machine (SVM). Zhou et al., [14] employed SVM, but they also specified local and global features on questions, user history, and question-user relationships, as well as taking Kullback-Leible divergence into consideration as a novel feature. Ji et al., [15] used text similarity as a feature when training RankingSVM, a version of SVM. For classifying the experts, additional techniques like random forest [16, 17] and Naive Bayes [18] are also used.

Language Models: By computing the word-based relevance of a user's prior behaviors to the query, language models compute the likelihood that a user would provide an answer to the question [19]. It is anticipated in a language model that the individuals whose profiles are most likely to generate the given query will also be the individuals who are

Table 1 Overview of the question routing methods

Method	Description	Pros	Cons
Classification Method	Classifies users as experts or non-experts based on a variety of features	Flexible, can handle a variety of features	Requires hand-crafted feature extraction
Language Model	Computes the likelihood that a user would provide an answer to a question based on their prior behaviors	Fast and efficient, easy to interpret	Ignores semantics, may not be accurate for long or complex questions
Topic Model	Measures relationships between words in a topic space rather than the word space	Can handle long and complex questions, can capture semantic relationships between words	Computationally expensive, requires all of the questions and answers data
Network-Based Method	Assesses users' authoritativeness in a user-user network created by their asking-answering relationships	Simple and easy to implement, can handle dynamic environments	Unable to take into account questions, themes, or categories
Collaborative Filtering Method	Uses user ratings of items to recommend items to other users	Flexible can handle a variety of features	May not be accurate for new users or items, may suffer from data sparsity

most likely to deliver an answer. Finally, the model offers a sorted list of users who are most likely to reply to the inquiry. Some variants of the language models are relevance-based language model [20], cluster-based language model [21], and hierarchical-based language model [22].

Topic Model: Later, topic models were developed that do not require the precise word to be in the user profile but instead measure relationships in the topic space rather than the word space [23]. Latent Dirichlet Allocation (LDA) [24] and Probabilistic Latent Semantic Analysis (PLSA) [25] are a couple of topic models that are frequently employed. In order to represent a document in a low-dimension space, Latent Semantic Indexing (LSI) [26], which is the foundation of PLSA, requires Singular Value Decomposition. The data creation process is modeled as a Bayesian network in PLSA using latent topics to represent documents [27, 28]. In the LDA approach, the topic mixture is derived from a conjugate Dirichlet prior that is constant across all users. First, topics based on historical user activity are extracted using LDA to demonstrate the relationship between knowledgeable people and fresh questions. These topics are used to calculate each user's likelihood of providing an answer in the second phase, and users are then ranked according to this likelihood [29].

Network-Based Method: The users with the highest levels of authority are suggested as the subject matter experts for new questions by the network-based approaches, which assess users' authoritativeness in a user-user network created by their asking-answering relationships. By employing the degree centrality measure InDegree [30], which values users who have answered more questions in the user-user network as better answerers, it is the most straightforward way to assess a user's authority in the CQA community. There are three basic network-based methods: PageRank [31], Hyperlink Induced Topic Search (HITS) [32], and Expertise Rank [30].

Collaborative Filtering Method: Matrix Factorization (MF) techniques, which are well known to be favorable in terms of flexibility and scalability in the recommendation domain, were employed in a different area of study [33]. In order to determine the user's level of competence on particular words, Zhao et al., [7] employed MF to represent the questions with the words that made up their content. The performance of the MF technique is, however, negatively impacted by the high-dimensional, sparse matrix that comes from this [34]. The semantic similarity between words is also disregarded because the matrix factorization approach treats each element as an independent entity.Following in this approach, Yang et al., [35] suggested performing the MF for question retrieval using tags, which condense the topic of the question rather than the text of the questions and responses. The study showed that tags are more useful and do a better job of summarising the subject of the query. The method also assesses the answerer's proficiency in a particular question and associated tags based on the number of votes for the answer. However, even if using tags rather than words somewhat resolves the dimensionality issue, it does not fully address the issue of data sparsity. The technique also continues to be hampered by the presence of a large number of associated items.To address one of the problems caused by the spelling variations of tagged keywords [35], Fukui et al., [8] attempted to enhance the strategy by expanding tagged keywords based on word embeddings.

There are a few drawbacks to the methods, including hand-crafted feature extraction required by the classification methodology. Word matching serves as the foundation of the language model, which leaves out semantics. Since topic extraction requires all of the questions and answers data, topic models in dynamic environments are computationally expensive. The network-based approach is unable to take into account the questions, themes, or categories that can be leveraged from the text. In conclusion, despite the fact that numerous innovative approaches to question routing have been proposed over the years, there are still a number of unresolved issues and potential areas for development, both of which this work aims to address.The table 1 provides a summary of the different question routing models.

3 Problem formulation

A static archive of a CQA website that preserves all question-and-answer sessions that have amassed through time serves as the foundation for a CQA Network. The CQA is made up of users set $U = u_1, u_2, \ldots u_m$, where m is the total number of users on the community website, and these users include both the asker and the expert. An expert is someone who possesses the knowledge necessary to respond to a specific question in their domain. Additionally, it includes the question set $Q = q_1, q_2, \ldots q_n$, where n represents the overall number of questions on the community website. The questionId, askerId, creation date, answerId, answererId, acceptedAnswerId, acceptedAnswererId, tag, title-body, bounty tuple is the format for each question $q_i \in Q$. The tuple for each $u_i \in U$ includes score, tag, questionId, answerId, and datePosted. As a result, the task of a recommendation system can be properly defined as having a user or expert predict a given query. Therefore, the structured approach of the recommendation system's mission is the given a question/user, forecast the experts/question respectively.

4 Proposed approach

The proposed SSC-CF expert recommendation system is depicted in Fig. 1. The recommended approach tries to generate scalable and accurate expert recommendations. The approach is defined in terms of two main parts. The construction of the recommendation models occurs in the first stage. The clustering of the voting score, dimensionality reduction using SVD++, and creation of similarity matrices for the items and users are some of the important tasks carried out in this phase. Using K-means++ with the elbow method algorithm, users are initially clustered according to their voting scores for questions, expertise, and temporal knowledge, while questions are grouped according to their tags, bodies, and titles. Then, using BERT embeddings with cosine similarity, semantic similarity is determined for each cluster. While processing, each cluster is run through SVD++ to obtain the decomposition matrices. Users and questions each have unique SVD++ models. The prediction and expert recommendation tasks are made for a specific question in the second phase after the first phase trains on the models constructed. A prioritized list of experts is actually provided by the recommender system in answer to the target question. To do this, the target question is presented to one of the clusters chosen in the first round. On the basis of the prior vote score, the SVD++ algorithm is then used to determine how similar the target expert is to the other experts. Finally, we combine user and question-based predictions using a weighted method.

The algorithm 1 provides a comprehensive set of steps to take to achieve expert prediction. This section provides a thorough overview of each step in the process.

Algorithm 1 Algorithm for the Proposed Work SSC-CF

Input : $DataVector\{U_m\}_1^M, DataVector\{Q_n\}_1^N$,KMeans++with Elbow Method

Output : $ListofRecommendedExperts$

1: User_Clustering=KMeans++ with Elbow Method($\{U_m\}_1^M, O_1$)
2: Question_Clustering=KMeans++ with Elbow Method($\{Q_n\}_1^N, O_2$)
3: User_Voting = Building the rating matrix for each user cluster from the available voting dataset of each user
4: Question_Voting = Building the rating matrix for each question cluster from the available voting dataset of available users
5: User_CF=User based SVD++ calculation in each cluster of User_Voting
6: Question_SVD++=Question based SVD++ calculation in each cluster of Question_Voting
7: Semantic_Simialrity=Calculating the semantic similarity between the questions in each question cluster.
8: Question_CF = Join(Question_SVD++,Semantic_Similarity)
9: Prediction_of_Expert = Combined_Outputs(User_CF,Question_CF)

4.1 Clustering module

Using the K-means++ with elbow approach, the user and question profiles are first clustered.The K-means++ algorithm is an improvement on the original K-means algorithm that aims to select the initial cluster centres more successfully. The standard K-means method selects the initial cluster centres at random, which might lead to less-than-ideal clustering results. By employing a more intricate initialization process that takes the distance between data points into consideration, K-means++ chooses the initial cluster centres in a way that is more likely to result in superior clustering. This initialization technique generates clustering results with faster convergence and higher accuracy when compared to the traditional K-means algorithm. By minimizing the sum across

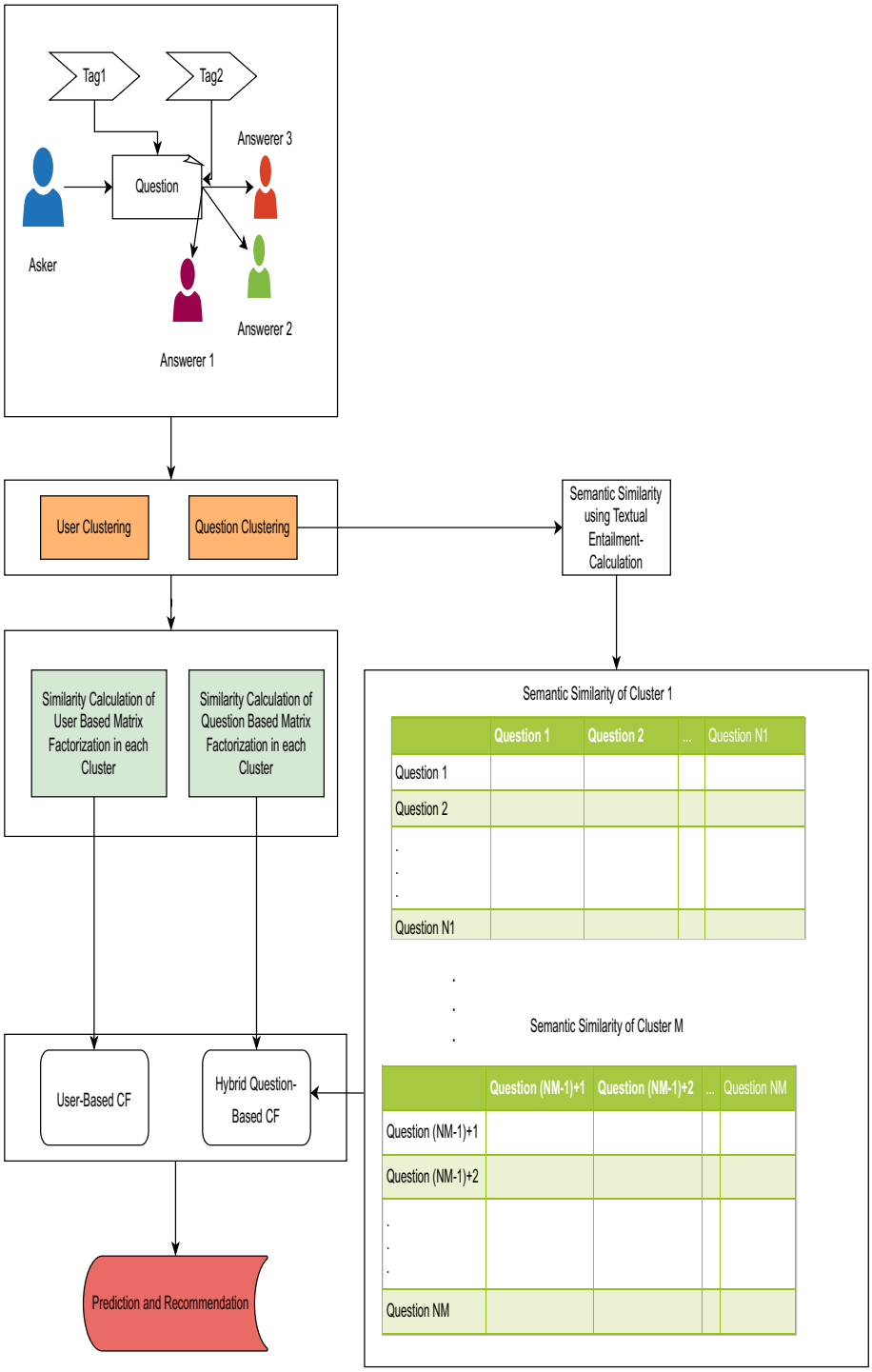

Fig. 1 Block Diagram of the Proposed Work SSC-CF

each cluster of the square of the distance between the point and its centroid, the K-Means++ with elbow technique algorithm seeks out k centroid positions $(C_1, C_2, \ldots C_k)$. The data are clustered using the Lloyd Algorithm, an iterative approximation algorithm. Both the efficient centroid initialization and the optimum number of clusters are employed in K-Means++ with the Elbow technique. The KMeans++ with Elbow Method is explained in detail in the algorithm 2. According to their voting results for questions, expertise, and temporal knowledge, users are grouped together based on similar preferences in the process of user clustering. The target user's prediction task is carried out using the aggregated opinions in each cluster after the clusters have been created. As a result, speed is improved because the cluster that needs to be studied contains far fewer people than the total number of users.

In Fig. 2, m represents the total number of users, a_{ij} represents the average vote score by user cluster center i for question j, R_{ij} represents the voting result for user i for question j, and n and c represents the total number of questions and user centers, respectively. Question clustering groups questions together according to similar tags, bodies, and titles. Following the clustering process, the target question is predicted using the aggregated answers to all other questions in any cluster. As a result, since the cluster that needs to be processed contains a lot fewer questions than all of them combined, performance is improved. In Fig. 3, m stands for the total number of users, a_{ij} is the average voting score for user i to question cluster center j, n for the total number of questions, R_{ij} for user i to question j's voting score, and k for the total number of question centers.

	Question 1	Question 2	. . .	Question n
User 1	R_{11}	R_{12}		R_{1n}
User 2	R_{21}	R_{22}		R_{2n}
. . .				
User m	R_{m1}	R_{m2}		R_{mn}

	Question 1	Question 2	. . .	Question n
Center 1	a_{11}	a_{12}		a_{1n}
Center 2	a_{21}	a_{22}		a_{2n}
. . .				
Center c	a_{c1}	a_{c2}		a_{cn}

Clustering Method

Fig. 2 User cluster in CF

	Question 1	Question 2	. . .	Question n
User 1	R_{11}	R_{12}		R_{1n}
User 2	R_{21}	R_{22}		R_{2n}
. . .				
User m	R_{m1}	R_{m2}		R_{mn}

	Center 1	Center 2	. . .	Center k
User 1	a_{11}	a_{12}		a_{1k}
User 2	a_{21}	a_{22}		a_{2k}
. . .				
User m	a_{m1}	a_{m2}		a_{mk}

Clustering Method

Fig. 3 Question cluster in CF

Algorithm 2 KMeans++ with Elbow Method

Input : $DataVectors\{x_n\}_1^N, Max\ number\ of\ clusters\ O$
Output : $ClusterswithCentroidandDataPoints$

procedure KMEANS++($DataVectors\{x_n\}_1^N, number\ of\ clusters\ K$)
  n ← RandomInteger(1,N)
  $\mu_1 \leftarrow x_n$
  for $k \leftarrow 2$ to K **do**
    for $n \leftarrow 1$ to N **do**
      $dist_n \leftarrow min_{i<k} \|x_n - \mu_i\|$
    end for
    for $n \leftarrow 1$ to N **do**
      $p_n \leftarrow dist_n^2 / \sum_i dist_i^2$
    end for
    n ← Discrete$(p_1, p_2, ..., p_N)$
    $\mu_k \leftarrow x_n$
  end for
  Return Centroid$\{\mu_k\}_{k=1}^K$
end procedure
procedure KMEANS($DataVectors\{x_n\}_1^N, number\ of\ clusters\ K$)
  $\{C_k\}_{k=1}^K \leftarrow KMeans++(\{x_n\}_1^N, K)$
  repeat
    for $n \leftarrow 1$ to N **do**
      Find the nearest Centroid For the data point x_n
      Assign the point to the cluster
    end for
    for $k \leftarrow 1$ to K **do**
      $C_k \leftarrow Mean\ of\ all\ points\ assigned\ to\ the\ cluster$
    end for
  until Convergence or a fixed number of iterations
  Return Clusters with Centroid and Data Points.
end procedure
procedure ELBOW_METHOD($DataVectors\{x_n\}_1^N, Max\ number\ of\ clusters\ O$)
  for $o \leftarrow 1$ to O **do**
    $Result\_Cluster_o \leftarrow$ KMeans($\{x_n\}_1^N, o$)
    $SSE_o \leftarrow \sum_{i=1}^o \sum_{x_j \in C_o} \|x_j - \mu_i\|^2$
  end for
  optimum ← index of SSE array with a minimum value
  Return optimum
end procedure
No_of_Cluster K=Elbow_Method($\{x_n\}_1^N$,))

$\{C_k\}_{k=1}^K \leftarrow KMeans++(\{x_n\}_1^N, K)$
repeat
  for $n \leftarrow 1$ to N **do**
    Find the nearest Centroid For the data point x_n
    Assign the point to the cluster
  end for
  for $k \leftarrow 1$ to K **do**
    $C_k \leftarrow Mean\ of\ all\ points\ assigned\ to\ the\ cluster$
  end for
until Convergence or a fixed number of iterations
Return Clusters with Centroid and Data Points.

4.2 SVD++

Then, in order to deal with predicting the unknowable values related to the sparsity, SVD++ is used in each cluster. The Singular Value Decomposition (SVD) algorithm has an extension known as SVD++ that is extensively used for matrix factorization and recommendation systems. A new term in the SVD++ technique takes into account implicit user input, such as how frequently or for how long a user engages with a certain item. With this improvement, SVD++ is more competent than the original SVD technique to express user preferences and offer recommendations. SVD++ is preferred over SVD for recommendation systems that take into account implicit user feedback. The SVD++ algorithm is an enhanced version of the classic SVD algorithm. It considers the user's voting score matrix R to be a product of two matrices, E and F. It also maps all users and all questions into a K-dimensional latent semantic space.

Table 2 Statistics of Stack Exchange Sites

	Gaming	Physics	Scifi
# Questions	75,696	93,529	38,026
# Answers	1,30,294	1,37,258	78,652
# Unique Users	51,192	41,115	26,673
# Questions having Best Answers	45,798	38,094	21,740
# Unique Tags	4,437	876	2,349
Avg # Tags per Question	1.2823	2.9634	2.1967
# Askers	25,153	31,415	12,413
# Asker (asked only 1 question) (%)	74.23%	63.26%	74.71%
Avg # Questions per Asker	2.9689	2.8849	3.0031

The semantic space is made up of a collection of latent elements. The user voting score matrix is factorized as follows:

$$R_{U*Q} = E_{U*K} * F_{K*Q} = \begin{bmatrix} e_{11}, & \dots & e_{1K} \\ e_{21}, & \dots & e_{2K} \\ & \dots & \\ e_{U1} & \dots & e_{UK} \end{bmatrix} * \begin{bmatrix} f_{11} & \dots & f_{1Q} \\ f_{21} & \dots & f_{2Q} \\ & \dots & \\ f_{K1} & \dots & f_{KQ} \end{bmatrix} \quad (1)$$

The user set is represented by $U = (u_1, u_2, \dots, u_n)$ and the question set is represented by $Q = (q_1, q_2, \dots, q_m)$. e_{ik} denotes the user i's expertise degree for the k-th latent factor in the question. The distribution of the k-th latent component is represented by f_{kj} among the questions j. As a result, each user has a user vector $e_u \in R$, which is a row of the matrix E. And each question has its own question vector $f_q \in R$, which is a column in the matrix F. e_u denotes the user's expertise. The question's feature space is described by f_q. They are now in the same data space. As a result, the usual collaborative filtering technique may be implemented here: the dot product of these two vectors is used to obtain the expert $\hat{r}_{uq}$ forecast, which is voted on by the user u for the question q. In the equation 2, the term $\hat{r}_{uq}$ is expressed.

$$\hat{r}_{uq} = e_u^T f_q = f_q^T e_u \quad (2)$$

Analyzing the user's voting score matrix R reveals that some users consistently give high or low voting scores compared to others. This suggests that the vote results are biased. However, equation 2 does not take bias into account. As a result, numerous bias factors need to be considered to obtain a more objective voting score. The Eqs. (3), (4), (5) is the voting score that has been modified.

$$\hat{r}_{uq} = b_{uq} + f_q^T e_u \quad (3)$$

$$b_{uq} = \mu + b_u + b_q \quad (4)$$

$$\hat{r}_{uq} = \mu + b_u + b_q + f_q^T e_u \quad (5)$$

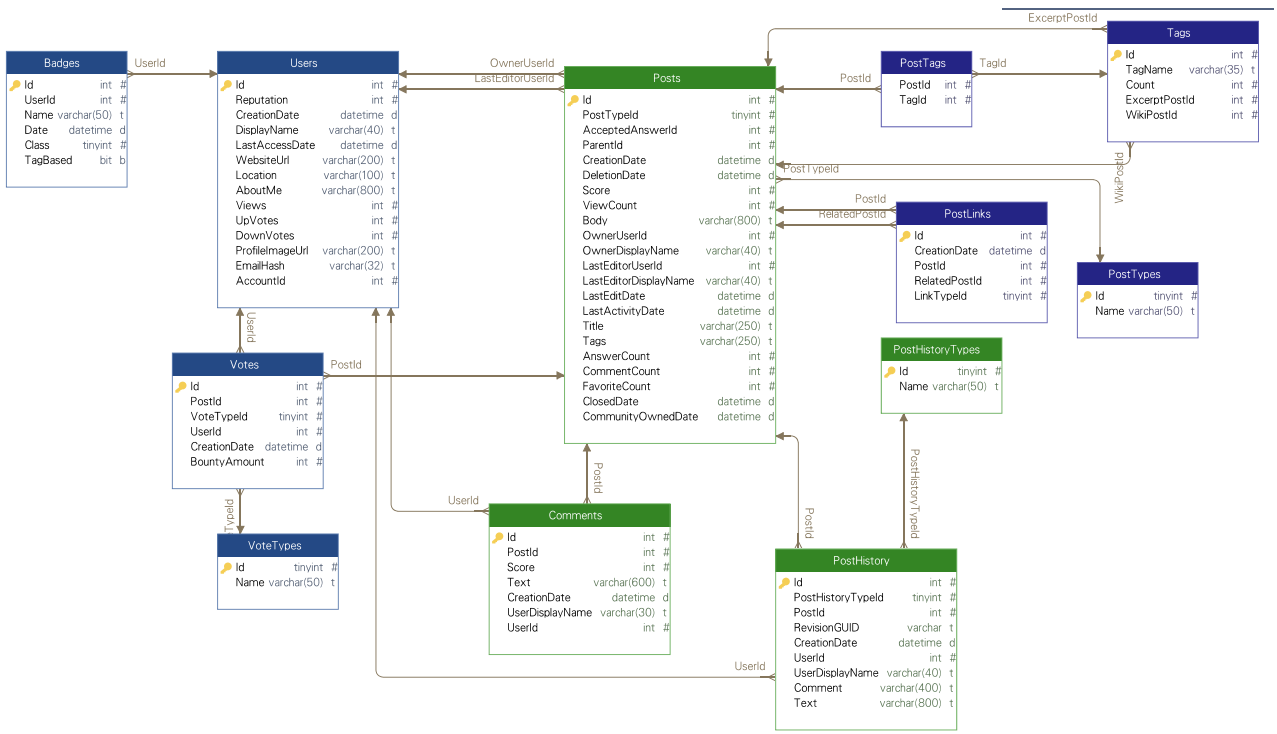

Fig. 4 Stack exchange sites database entity relationship

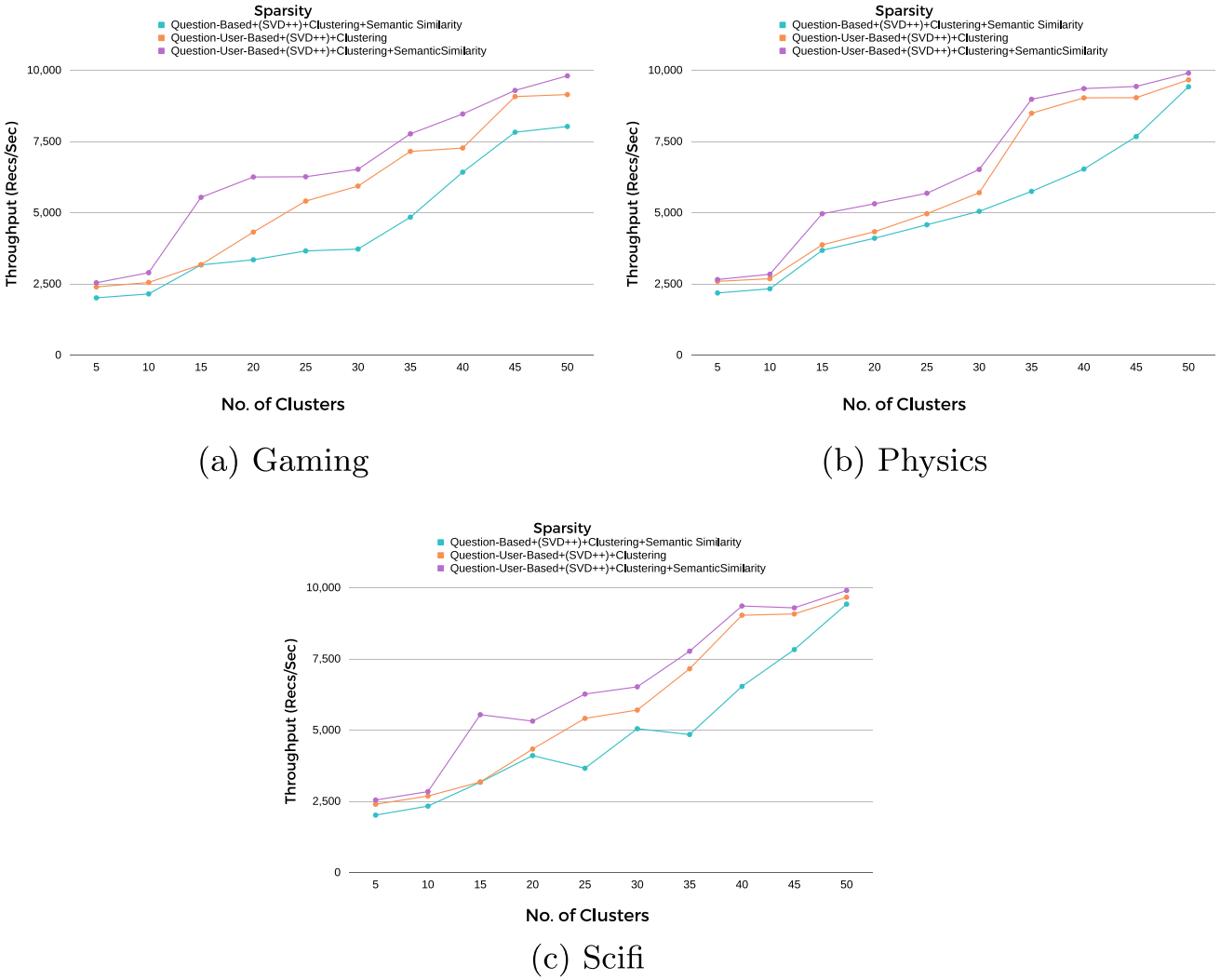

Fig. 5 Throughput

b_{uq} indicates the user u's total bias information for the question q, whereas μ is the voting score mean. The bias information of the question q is represented by b_q, which is an item offset to the voting score mean. The bias information of the user u, which is a user offset to the voting score mean, is represented by b_u. Apart from the biased information, many implicit parameters are added to SVD++ to better reflect the user's latent competence of the question. In general, users' voting scores are referred to as explicit information, whereas user behavior is referred to as implicit information. Finally, a user's preference perspective is obtained by merging the previously mentioned explicit information, bias information, and implicit information. The final voting score is given in the equation 6.

$$\hat{r}_{uq} = \mu + b_u + b_q + f_q^T \left(e_u + |N_u|^{\frac{-1}{2}} \sum_{j \in N_u} y_j \right) \tag{6}$$

N_u represents the user u's behavior data. This indicates that the user u voted on the corresponding question. The size of the behavior data is given by $|N|$. $-(\frac{1}{2})$ is an empirical value for the contraction factor. y_j represents the implicit parameters used in the recommendation to describe the implicit information, indicating that the user u voted for the question q. Finally, in addition to explicit information, the SVD++ algorithm considers both bias information and implicit information. It first examines users' levels of expertise for the question's latent semantic factors. The distribution of the latent semantic factors among all questions is then obtained. Finally, it considers both the bias information and the implicit information mentioned above. Based on the preceding analysis, the equation 7 represents the cost function J of the SVD++ method.

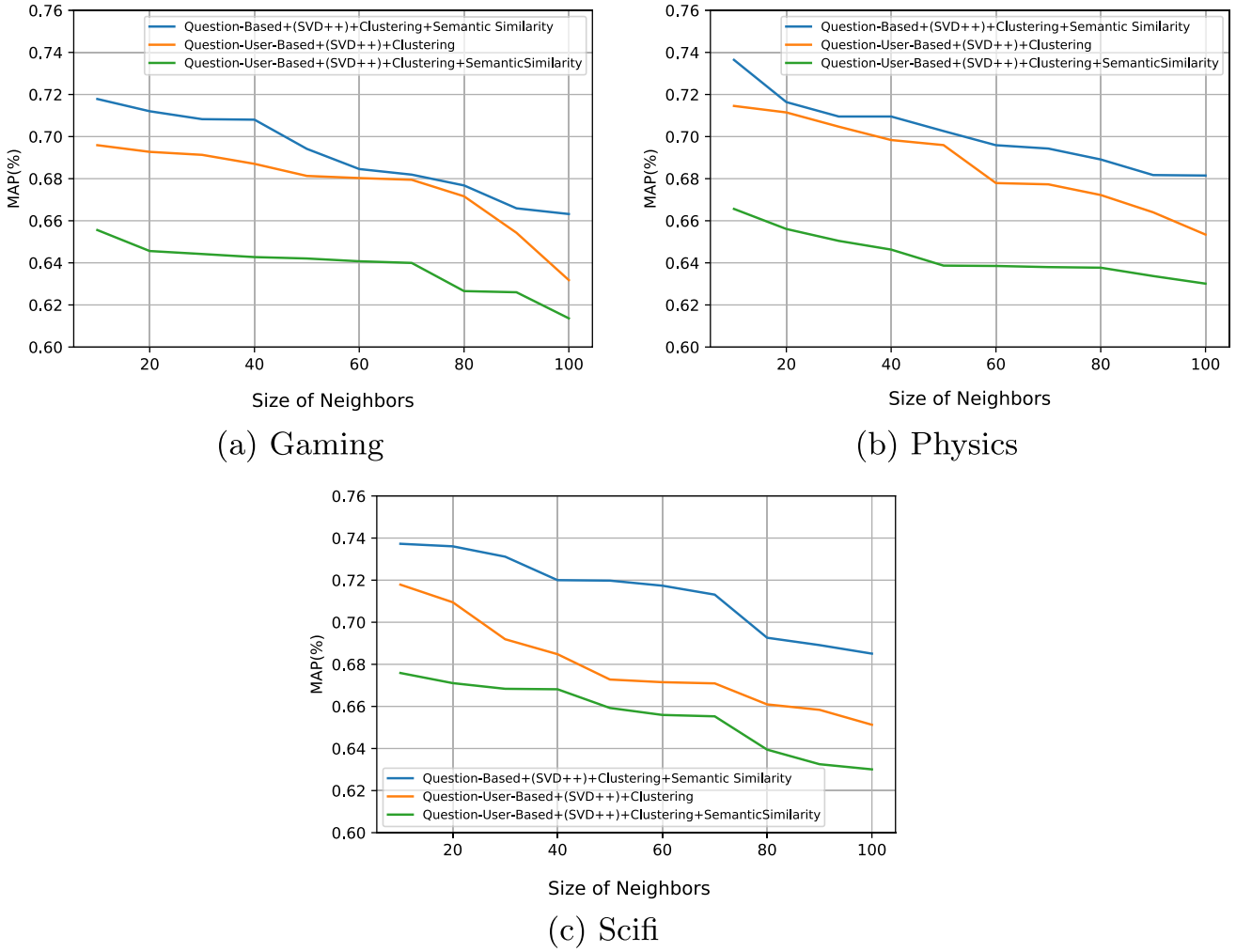

Fig. 6 MAE

Table 3 Precision for Different number of Top-N

No.of Clusters	Gaming			Physics			Scifi		
	Method A	Method B	Method C	Method A	Method B	Method C	Method A	Method B	Method C
Top-5	0.650315	0.707921	0.761291	0.650463	0.70224	0.753387	0.651525	0.700138	0.750754
Top-10	0.656755	0.710416	0.766574	0.650493	0.715552	0.754921	0.651929	0.743206	0.766614
Top-15	0.658563	0.718605	0.768485	0.657254	0.718493	0.75526	0.657521	0.748513	0.768185
Top-20	0.671018	0.726582	0.775988	0.674773	0.729018	0.758726	0.668499	0.751167	0.773769
Top-25	0.688533	0.730652	0.78033	0.702913	0.738001	0.769431	0.670109	0.752746	0.783726
Top-30	0.690591	0.75907	0.782991	0.711339	0.744611	0.775696	0.68002	0.755993	0.785393
Top-35	0.697544	0.771597	0.783855	0.720074	0.756747	0.788304	0.71112	0.756442	0.797364
Top-40	0.705417	0.78149	0.807137	0.721821	0.757149	0.799595	0.735993	0.758696	0.800258
Top-45	0.723071	0.785142	0.810861	0.729342	0.791069	0.806874	0.738865	0.788234	0.806764
Top-50	0.740147	0.791299	0.818101	0.747797	0.794814	0.818696	0.747023	0.794793	0.809041

Method A: question-user-based+(SVD++)+clustering

Method B: question-based+(SVD++)+clustering+semantic similarity

Method C: question-user-based+(SVD++)+clustering+semantic similarity

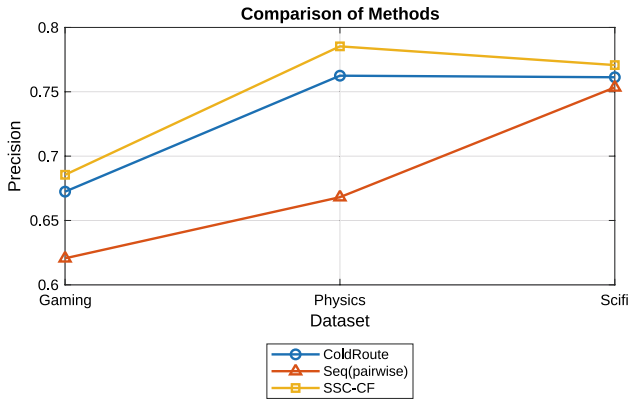

Fig. 7 Comparison of the proposed model SSC-CF with other approaches

$$J = \min_{b_q, b_u, f_q, e_u} \sum_{(u,e)\in K} \left(r_{uq} - \mu - b_u - b_q - f_q^T \left(e_u + |N_u|^{-1/2} \sum_{j\in N_u} y_j \right) \right)^2 + \lambda \left\{ \sum_u \left(b_u^2 + \|e_u\|^2 \right) + \sum_q \left(b_q^2 + \|f_q\|^2 + \|y_q\|^2 \right) \right\} \tag{7}$$

The first component of J is the loss calculated using the least square method. The regularisation term is the second part of J. The stochastic gradient descent method is used to optimize (equations (8), (9), (10), (11), (12), (13)) the proposed SVD++ algorithm:

$$b_u \longleftarrow b_u + \gamma \left(g_{uq} - \lambda b_u \right) \tag{8}$$

$$b_q \longleftarrow b_q + \gamma \left(g_{uq} - \lambda b_q \right) \tag{9}$$

$$e_u \longleftarrow e_u + \gamma \left(g_{uq} f_q - \lambda e_u \right) \tag{10}$$

$$f_q \longleftarrow f_q + \gamma \left(g_{uq} \left(e_u + |N_u|^{-1/2} \sum_{j\in N_u} y_j \right) - \lambda f_q \right) \tag{11}$$

$$y_j \longleftarrow y_j + \gamma \left(g_{uq} |N_u|^{-1/2} f_q - \lambda f_q \right) \tag{12}$$

$$g_{uq} = r_{uq} - \hat{r}_{uq} \tag{13}$$

The three variables are g_{uq} (prediction error), γ (learning rate), and λ (regularisation parameter).

4.3 Semantic similarity module

In order to further increase the predictive accuracy, semantic similarity between the questions is determined, and the final predictions are made using a hybrid of both SVD++ and semantic similarity in each cluster of the question profile. The proposed work SSC-CF uses BERT [36] with cosine distance to calculate the semantic similarity between the questions. Over time, word vectors have changed from a one-hot environment in which every word was orthogonal to every other word to one in which word vectors can change to fit the context. BERT has the capacity to incorporate word meaning into densely packed vectors. Each value contained in the dense vector has a value and a purpose for existing in that value. Each encode layer outputs a set of dense vectors thanks to BERT's prowess in producing them. The ability to obtain word vectors that morph based on context is made possible by language modeling models. The static embedding layer, the first layer of BERT's total of 13 layers, was selected for initial training. After obtaining the embeddings, the inputs to which the semantic similarity calculation must be applied are turned into a vector. The cosine similarity between the vectors is then used to calculate the semantic similarity. By computing the cosine of the angle created by two vectors projected in three dimensions, cosine similarity is computed. The cosine similarity between two vectors with the same orientation is 1, but the similarity between two vectors oriented at 90 degrees is 0. Equation 14 contains the mathematical formula for the cosine similarity of vectors A and B.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{14}$$

Finally, based on the user and hybrid-question CF, the expert in the CQA is predicted.

5 Experimental results

The analysis examines the suggested recommendation system, SSC-CF, using data from stack exchange sites. The dataset used for the experiment is described in Sect. 5.1. The performance of the recommended system SSC-CF is evaluated in the Sect. 5.2

5.1 Dataset description

Data from the three exchange sites are used in this study. The table 2 contains some statistics on the data from stack exchange sites. The data is extracted from the stack exchange

sites' dump.[4] Fig. 4 displays the entity relationship diagram for the dataset being evaluated from stack exchange platforms.

5.2 Evaluation

The SSC-CF method combines the SVD++ and semantic similarity measures with question- and user-based clustering. Comparing the proposed technique SSC-CF to the question-based+(SVD++)+clustering+semantic similarity and question-user-based+(SVD++)+clustering recommendation systems. SSC-CF is examined for throughput, which is referred to as the number of suggestions per second, on the three stack exchange sites to demonstrate the efficacy of the suggested solution in addressing the scalability issue. The performance outcomes of all methods experiments are shown in the Fig. 5. The methods throughput is graphed as a function of cluster size. The clustering process in the Proposed Approach SSC-CF employs the K-means++ with elbow method. For the suggested method SSC-CF, various clustering sizes are taken into account. Plots show that methods that use clustering, dimensionality reduction, and semantic similarity techniques have significantly higher throughput than other methods. Additionally, it can be shown from the Fig. 5 that the throughput of the approaches increases with an increase in clustering size. The Mean Absolute Error (MAE) between the projected and actual vote score is assessed using statistical measures. In contrast, decision-support metrics, for example, by measuring the overlap, compare the suggested items with the pertinent ones. Equation 15 presents MAE.

$$\text{MAE (pred, act)} = \sum_{i=1}^{N} \left| \frac{\text{pred}_{u,i} - \text{act}_{u,i}}{N} \right| \tag{15}$$

where N is how many questions a person has taken and received a score on. The MAE is used to assess the predictive accuracy of the SSC-CF approach. For various neighboring densities, it has been evaluated. The prediction accuracy for various neighborhood sizes on datasets is shown in Fig. 6. For the multi-criteria recommender assessments in terms of accuracy measures, decision-support metrics, in particular, will be crucial. The following discussion will cover the measures for this use that are well-known in the field of information retrieval. The precision Eq. 16 counts the number of relevant items in the received result.

$$\text{Precision} = \frac{TR}{TR + FR} \tag{16}$$

where FR stands for false relevant forecasts, TR stands for true relevant predictions, and FN stands for false non-related predictions. The precision is calculated on several Top-N numbers in order to evaluate the suggested method using decision-support accuracy metrics. In this study, N = 10, 20, 30, 40, and 50 are taken into account, meaning that we analyze the approach when recommending the top 10, 20, 30, 40, and 50 movies using the suggested recommender system. Table 3 displays the precision figures for various Top-N. The table shows that our novel method produced rather high-precision results. The metrics show that the proposed method performs better than the alternative methods. Likewise, the precision of the approach suggested by Sun et al., [37] is outperformed by the proposed SSC-CF method and the Fig. 7 giveses us the comparison between the model SSC-CF, ColdRoute [37] and Seq(Pairwise) [38]. These findings are enough to back up our contention that our strategy is relatively scalable and improves accuracy. Two major problems with recommender system design are scalability and sparsity. In order to improve the functionality of recommender systems, efforts have been made to address these problems in this research. With the use of clustering and dimensionality reduction techniques, the method created in this work takes advantage of semantic similarity in the SSC-CF. Three datasets from stack exchange sites were utilized to evaluate the approach. According to the findings presented by MAE and Precision, the performance of the CF recommender systems was enhanced by the application of semantic similarity in conjunction with clustering and dimensionality reduction approaches. The analysis's findings showed that the scalability and sparsity problems in recommender systems can be resolved using the hybrid recommendation method.

6 Conclusion

The proliferation of community forums has recently increased the importance of tasks associated with them. It can be challenging to identify whom to look for answers to because of the constant influx of new questions on these forums. The issue of questions with no answers is attempted to be resolved by recommending inquiries to experts. The study focused on the collaborative filtering approach's cold start issue and data sparsity issue for the expert recommendation system. The SSC-CF is a proposed approach that employs dimensionality reduction to deal with scalability or the cold-start problem and semantic similarity to boost accuracy. The experiments make use of the three most well-known three stack exchange platforms dataset such as gaming, physics, scifi. When the metrics were used to evaluate the SSC-CF, the findings showed that the proposed technique, SSC-CF, performed better due to the help of semantic

[4] https://archive.org/details/stackexchange.

similarity in addition to clustering and dimensionality reduction. Despite the fact that the study suggests a strategy for dealing with scalability and data sparsity issues, future work will focus on increasing the accuracy of the results obtained.

Author Contributions A.P. and S.N. equally contribute towards the idea, experiments, interpretation of results and paper write-up.

Funding This work received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability The datasets analysed during the current study are available in the Stack Exchange Data Dump https://archive.org/details/stackexchange

Declarations

Conflict of interest The authors report there are no competing interests to declare.

Ethical Approval Not Applicable

References

1. Yuan S, Zhang Y, Tang J, Hall W, Cabotà JB (2020) Expert finding in community question answering: a review. Artif Intell Rev. 53(2):843–874. https://doi.org/10.1007/s10462-018-09680-6
2. Faisal MS, Daud A, Akram AU, Abbasi RA, Aljohani NR, Mehmood I (2019) Expert ranking techniques for online rated forums. Comput Human Behav. 100:168–176. https://doi.org/10.1016/j.chb.2018.06.013
3. Choudhury SS, Mohanty SN, Jagadev AK (2021) Multimodal trust based recommender system with machine learning approaches for movie recommendation. Int J Inf Technol. 13:475–482. https://doi.org/10.1007/s41870-020-00553-2
4. Jena KK, Bhoi SK, Mallick C, Jena SR, Kumar R, Long HV et al (2022) Neural model based collaborative filtering for movie recommendation system. Int J Inf Technol. 14(4):2067–2077. https://doi.org/10.1007/s41870-022-00858-4
5. Saraswat M, Srishti (2022) Leveraging genre classification with RNN for Book recommendation. Int J Inf Technol. 14(7):3751–3756. https://doi.org/10.1007/s41870-022-00937-6
6. Tareq SU, Noor MH, Bepery C (2020) Framework of dynamic recommendation system for e-shopping. Int J Inf Technol. 12(1):135–140. https://doi.org/10.1007/s41870-019-00388-6
7. Zhao Z, Zhang L, He X, Ng W (2014) Expert finding for question answering via graph regularized matrix completion. IEEE Trans Knowl Data Eng. 27(4):993–1004. https://doi.org/10.1109/TKDE.2014.2356461
8. Fukui K, Miyazaki T, Ohira M (2019) Suggesting Questions that Match Each User's Expertise in Community Question and Answering Services. In: 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). IEEE. p. 501–506
9. Sohail SS, Siddiqui J, Ali R. Classifications of recommender systems: A review. J Eng Sci Technol Rev. 2017;10(4). doi: https://doi.org/10.25103/jestr.104.18
10. Kumar P, Thakur RS (2018) Recommendation system techniques and related issues: a survey. Int J Inf Technol. 10:495–501. https://doi.org/10.1007/s41870-018-0138-8
11. Najafabadi MK, Mohamed A, Onn CW (2019) An impact of time and item influencer in collaborative filtering recommendations using graph-based model. Inf Process Manag. 56(3):526–540. https://doi.org/10.1016/j.ipm.2018.12.007
12. Koren Y (2010) Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. ACM Trans Knowl Disc Data. 4(1):1–24. https://doi.org/10.1145/1644873.1644874
13. Neshati M, Fallahnejad Z, Beigy H (2017) On dynamicity of expert finding in community question answering. Inf Process Manag. 53(5):1026–1042. https://doi.org/10.1016/j.ipm.2017.04.002
14. Zhou TC, Lyu MR, King I (2012) A classification-based approach to question routing in community question answering. In: Proceedings of the 21st international conference on world wide web; p. 783–790
15. Ji Z, Wang B (2013) Learning to rank for question routing in community question answering. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. p. 2363–2368
16. Choetkiertikul M, Avery D, Dam HK, Tran T, Ghose A (2015) Who will answer my question on stack overflow?. In: 24th Australasian Software Engineering Conference. IEEE p. 155–164
17. Sorkhani S, Etemadi R, Bigdeli A, Zihayat M, Bagheri E (2022) Feature-based question routing in community question answering platforms. Inf Sci. 608:696–717. https://doi.org/10.1016/j.ins.2022.06.072
18. Van Dijk D, Tsagkias M, De Rijke M (2015) Early detection of topical expertise in community question answering. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. p. 995–998
19. Zhou G, Liu K, Zhao J (2012) Joint relevance and answer quality learning for question routing in community qa. In: Proceedings of the 21st ACM international conference on Information and knowledge management. p. 1492–1496
20. Lavrenko V, Croft WB (2017) Relevance-based language models. ACM SIGIR Forum, vol 51. ACM New York, NY, USA, pp 260–267
21. Liu X, Croft WB (2004) Cluster-based retrieval using language models. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. p. 186–193
22. Petkova D, Croft WB (2008) Hierarchical language models for expert finding in enterprise corpora. Int J Artif Intell Tools. 17(01):5–18. https://doi.org/10.1109/ICTAI.2006.63
23. Riahi F, Zolaktaf Z, Shafiei M, Milios E. Finding expert users in community question answering. In: Proceedings of the 21st international conference on world wide web; 2012. p. 791–798
24. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res. 3:993–1022. https://doi.org/10.5555/944919.944937
25. Hofmann T. Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval; 1999. p. 50–57
26. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci. 41(6):391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9
27. Wu H, Wang Y, Cheng X. Incremental probabilistic latent semantic analysis for automatic question recommendation. In: Proceedings of the 2008 ACM conference on Recommender systems; 2008. p. 99–106
28. Qu M, Qiu G, He X, Zhang C, Wu H, Bu J, et al. Probabilistic question recommendation for question answering communities. In: Proceedings of the 18th international conference on World wide web; 2009. p. 1229–1230

29. Momtazi S, Naumann F (2013) Topic modeling for expert finding using latent Dirichlet allocation. Wiley Interdiscip Rev Data Min Knowl Discovery. 3(5):346–353. https://doi.org/10.1002/widm.1102
30. Zhang J, Ackerman MS, Adamic L (2007) Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th international conference on World Wide Web. p. 221–230
31. Borodin A, Roberts GO, Rosenthal JS, Tsaparas P (2005) Link analysis ranking: algorithms, theory, and experiments. ACM Trans Internet Technol. 5(1):231–297. https://doi.org/10.1145/1052934.1052942
32. Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. J ACM (JACM). 46(5):604–632. https://doi.org/10.1145/324133.324140
33. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer. 42(8):30–37. https://doi.org/10.1109/MC.2009.263
34. Idrissi N, Zellou A (2020) A systematic literature review of sparsity issues in recommender systems. Social Netw Anal Min. 10(1):1–23. https://doi.org/10.1007/s13278-020-0626-2
35. Yang B, Manandhar S (2014) Tag-based expert recommendation in community question answering. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). IEEE. p. 960–963
36. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018;
37. Sun J, Vishnu A, Chakrabarti A, Siegel C, Parthasarathy S (2018) Coldroute: effective routing of cold questions in stack exchange sites. Data mining and knowledge discovery. 32(5):1339–1367. https://doi.org/10.1007/s10618-018-0577-7
38. Sun J, Zhao J, Sun H, Parthasarathy S (2021) EndCold: An end-to-end framework for cold question routing in community question answering services. In: Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence. p. 3244–3250