A novel framework for MOOC recommendation using sentiment analysis

Sujatha Uthamaraj[1], Gunasundari Ranganathan[2]
[1]Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India
[2]Department of Computer Applications, Karpagam Academy of Higher Education, Coimbatore, India

Article Info

Article history:

Received Feb 29, 2024
Revised Jun 13, 2024
Accepted Jun 25, 2024

Keywords:

Chatbot
Deep learning
Hybrid recommendation
Sentiment analysis
Web scraping

ABSTRACT

Massive open online courses (MOOC) are the largest initiative in eLearning, with the support of universities across the world. To increase course satisfaction in MOOCs, learners' must relate to the courses that best suit their needs and interests. The goal of recommendation systems is to suggest items to users based on their preferences and past behaviour. A course recommender system makes recommendations based on the similarity of courses and past interactions with the MOOC platform. With a huge volume of online courses on multiple learning platforms, it has been difficult for learners to identify the course of their interest. To address these challenges, a novel framework for hybrid MOOC course recommendations is proposed to recommend courses from multiple learning platforms. It uses web scraping techniques to collect course data from various MOOC providers, such as Coursera, Udemy, and edX platforms. With the real time dataset, a deep learning chatbot captures the personalized learning requirements of learners and recommends using a user-user collaborative approach with the valence aware dictionary and sentiment reasoner (VADER) for sentiment analysis. It enhances the accuracy of recommendations with an root-mean-square error (RMSE) value of 0.541.

Corresponding Author:

Sujatha Uthamaraj
Department of Computer Science, Karpagam Academy of Higher Education Coimbatore
Tamil Nadu, India
Email: sujathau@gmail.com

1. INTRODUCTION

Massive open online courses (MOOCs) have gained widespread popularity after COVID-19. Learners have the opportunity to choose from a variety of MOOC offerings on the numerous learning platforms, tailoring their selections to align with specific learning interests. Users encounter substantial challenges while trying to identify the most relevant courses to meet their learning needs [1]. As new courses are added to MOOC sites day by day, the recommendation system requires a real-time data collection method. Most recommendation systems use high course ratings as the key element for recommendation, not considering the reviews of users. To bridge the gap, there is a need for a hybrid framework that consists of a web scraping technique to collect course information from MOOC platforms and a chatbot to identify the personalized learning requirements with a recommendation algorithm that not only considers high ratings of courses but also analyzes the sentiment of the reviews in finding suitable courses.

The recommendation systems can be categorized into collaborative recommendation, content-based recommendation, and a combination of both hybrid recommendations. In the initial recommendation models, content-based recommendations with similarity attributes like the pearson correlation coefficient and cosine similarity are considered. It relies on identifying the user's chosen attributes and provides recommendations using the similarity of courses [2]. To identify similarities between courses using the term frequency-inverse document frequency (TF-IDF), Word2Vector, and Elman minimal redundancy models, Vedavathi and Kumar [3], developed a robust content-based course recommendation with an accuracy of 99.98% and a recall of 99.81%. Collaborative recommendations are utilized to assess interactions between courses and students. It recommends courses with input from the user's past behaviour or the reviews associated with a particular recommended item. By accessing the learner's profile, it provides recommendations based on information about the learner's age, country, educational background, and past learning experiences. When a new learner enters the platform, they suffer from a cold start issue [4]. A collaborative-based personal news recommendation by Han [5], applied dichotomous K-means algorithm to recommend news with improved accuracy of 96.5% and recall rates of 72%.

The proposed hybrid approach aims to address the challenges of "cold start" that are often encountered by the other recommendation methods [6]. A hybrid recommender system represents a fusion of various techniques that utilize diverse inputs for the recommendation service, with each technique systematically complementing the others. It enhances the quality of recommendations and overcomes the limitations of content and collaborative recommendation techniques [7]. Afoudi *et al.* [8], integrate a hybrid system that combines collaboration with content-based recommendation using the self-organizing map neural network technique and is evaluated using a movie database with an RMSE value of 0.556. A recommendation system, MoodleREC by De Medio *et al.* [9], identifies the learning requirements to support course creation on the MOOC platform.

A substantial amount of research was conducted independently to apply content-based, collaborative, and hybrid recommendations for individual MOOC sites with the available dataset [10]. To the best of our knowledge, there is currently no other hybrid framework for course recommendation with real-time course information collection using web scraping in the initial phase, user-personalized requirements capturing through a deep learning chatbot in the following phase and integrating a hybrid recommendation with sentiment analysis is available. This novel framework integrates adjusted cosine similarity in the content-based approach and user-user collaborative filtering with sentiment analysis [11] to create a hybrid recommendation system that offers personalized, sentiment-aware recommendations to users. The research objectives are as follows:

- To create dataset with real time course information from MOOC sites through web scraping using Selenium and BeautifulSoup.
- This work involves the development of a deep learning chatbot to receive users' learning requirements.
- To develop a personalized hybrid recommendation algorithm using user-user collaborative filtering with adjusted cosine similarity for course recommendation with VADER sentiment analysis.

The remainder of this paper is organized as follows: section 2 provides a description of the methods used in the hybrid recommendation framework, and the algorithm for making recommendations is presented. The description of dataset, experimental results, discussion, and limitations will be presented in section 3. The conclusion and ideas for future works will be presented in section 4.

2. METHOD

2.1. Block diagram

Within an e-learning context, the hybrid framework provides recommendations for selecting courses that are relevant to their learning objectives [12]. With the large volume of MOOC courses, users often struggle to identify the right type of course. A novel framework, shown in Figure 1 is proposed for personalized hybrid course recommendation with sentiment analysis. The framework contains methods to capture user requirements through a chatbot, pre-processing the input, web scraping techniques, and a recommendation algorithm with sentiment analysis to recommend courses as shown in Figure 1(a). In the initial phase, deep learning chatbot model is trained to respond to user queries, and from the web-scraped course dataset, with fields the course name, level of the course, course URL, and ratings of the course are extracted, course reviews are extracted from the Kaggle course review dataset [13]. Sentiment analysis with VADER, a lexicon and rule-based sentiment analysis tool used to analyze the sentiments concealed within the review data and determine the sentiment scores of user reviews [14], [15]. The course dataset and the review dataset with sentiment scores are used for recommendation by a content-based and user-based collaborative approach. The workflow of these methods, as shown in Figure 1(b), helps in enhancing accuracy and predicting the top N recommended courses [16].

2.2. Chatbot

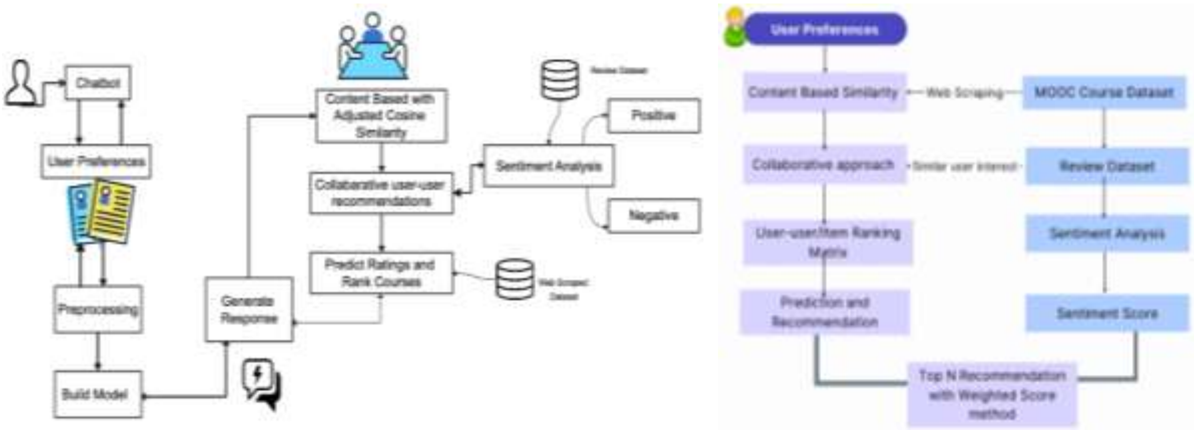
Chatbots are the traditional method for interacting with computers using natural language. A chatbot is a computer program designed to emulate the conversational manner of a real person, engaging in intelligent discussions with one or more individuals in human-like language [17]. Typically, chatbots utilize simple text interfaces for communicating with their human counterparts [18], although some also incorporate speech recognition and text-to-speech features [19]. Tzeng *et al.* [20], developed a LINE, a social media chatbot developed using the Markov decision process, was utilized to promote MOOC participation on the National Tsing Hua University MOOC platform. In the proposed work, to capture learners' choices of courses across MOOC platforms, a chatbot was developed to identify personal preferences and become the user interface for displaying recommendations. Chatbots are built using a deep learning model with multiple hidden layers and an output layer that contains a number of neurons equal to the number of intents and predicts output intent with the SoftMax function.

The process of creating a chatbot involves many steps: identifying the data source, cleaning and pre-processing the data to remove noise and irrelevant information. The data needed to be pre-processed before analysis. This involved the removal of various elements such as noise, punctuation, stop-words, URLs, and numerical values. The natural language toolkit (NLTK) library in Python is used for natural language processing (NLP). NLTK provides a wide range of text corpora, lexicons, and models for text processing [21]. Figure 2 explains the functions of chatbot, when a user interacts with the chatbot, the text undergoes many processes before getting a reply in its interface. Figure 2(a) explains that the flow of data pre-processing, it starts with tokenization, which involves breaking down text into individual words or tokens. With NLP, common stop words that can be excluded to emphasize meaningful words. Additionally, it facilitates stemming and lemmatization to streamline words to their base or root form and tag words with their parts of speech and remove punctuation. The chatbot model takes pre-processed user inputs match against predefined intent categories for response generation, as given in Figure 2(b). A dialogue corpus that comprises a wide range of intents and responses acts as the training data for the chatbot.

2.3. Web scraping

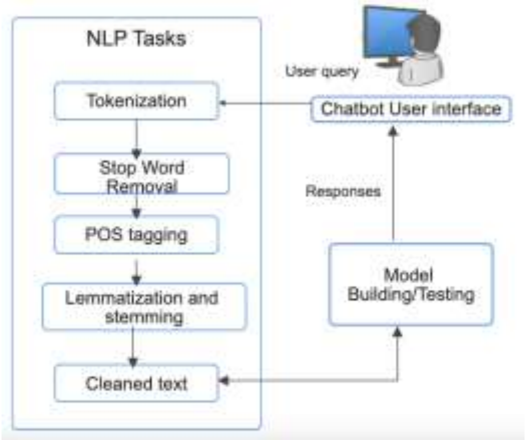
The role of datasets is pivotal in the advancement of deep learning models, and any model's construction relies on the availability of the relevant datasets. To accomplish the research objectives, it was essential to collect real-time course information from different MOOC providers through web scraping. Web scraping is a technology to extract information from websites in an automated manner and offers structured data. Considering the multitude of MOOC platforms that offer a wide array of courses, the investigations were limited to the Coursera, edX, and Udemy platforms. Python with the Selenium tool is utilized for web scraping [22] to gather course information from the MOOC sites. Figure 3 explains the web scraping process, starts with web site selection and inspection of HTML code on the required data to be collected. The beautiful soup library in Python parses HTML code and delivers text data. Data pre-processing on the text data generates a structured dataset used for data analysis.

Installing Selenium and Google Chrome is a prerequisite to access the web page. After the website has been identified, web scraping uses elements such as html tag name, class name, IDs, XPath, and CSS, for identifying course name, university, description, skills, course URL, level, duration, rating, specialization, language, subtitles, and specialization URL from the web pages. After pre-processing, 13,689 course information stored in the structured course dataset. Web scraping guarantees access to immediate and accurate data for recommendations.

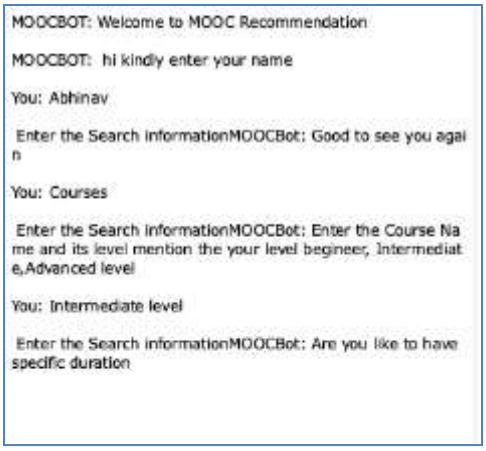


(a) (b)

Figure 1. Hybrid course recommendation (a) framework and (b) workflow



(a)



(b)

Figure 2. Working of Chatbot (a) process flow and (b) user interface

Figure 3. Web scraping technique

2.4. Hybrid recommendation

The hybrid recommendation approach offers a comprehensive, adaptive solution for MOOC platforms that balances personalization and diversity to meet the needs of a diverse user base. By leveraging the powers of both content and collaborative methods, hybrid systems [23] aim to enhance the learner's experience and engagement with the MOOC platform. It satisfies a distinct range of learners with varying preferences, backgrounds, and learning goals in a wide array of courses. New users joining the platform or new courses with limited historical data often face the "cold-start" problem [24], where collaborative filtering may not work well due to the lack of user-item interactions. Ezaldeen *et al.* [14] proposed a novel e-learning hybrid recommender system framework to recommend contents according to the learning needs using convolutional neural network (CNN).

Considering the advantages of previous works, a novel hybrid recommendation system developed with adaption to individual user behaviours and preferences over time. To improve the accuracy of recommendation, user-user based recommendation method is preferred in this work. The proposed hybrid method given in Figure 4 employs cascade approaches for performing course recommendations. Figure 4(a) depicts the work flow of hybrid approach with its algorithmic setup in Figure 4(b). In sentiment analysis, positive user reviews are used to predict the rating of new users with the preference of similar users to refine and personalize recommendations, thus resolving the cold start issue [25].

Content-based recommendation with TF-IDF with cosine similarity is utilized for suggesting courses to users based on the content and preferences of users. The objective of content-based filtering is to group courses sharing common attributes, considering the user's preferences, and subsequently search for these terms within the dataset [26]. Finally, recommend similar courses that share these common attributes using the TF-IDF and cosine similarity models. In formal terms, a recommender system is concerned with a set of users denoted as "U" and a set of courses denoted as "C." For each user-course pair (u_i, c_j), a recommender calculates a score or rank denoted as "r_{ij}" to gauge the anticipated interest of user ui in course c_j. The framework uses the ranking computation of courses using TF-IDF with cosine similarity. Cosine similarity is a measure of similarity between two non-zero vectors [27] in an inner product space. It is used to determine the similarity of two text documents based on the angles between their TF-IDF vectors. It computes the cosine of the angle between two vectors, yielding a value within the range of -1 (indicating complete dissimilarity) to 1 (indicating complete similarity). A value of 0 means the documents are orthogonal, indicating no similarity.

TF-IDF is a numerical model used in NLP to assess the significance of a term within a document. It is mathematically represented as in (1). Term frequency (TF) identifies the frequency of term (word) appears within a document.

$$TF(t,d) = \frac{f(t,d)}{|d|} \tag{1}$$

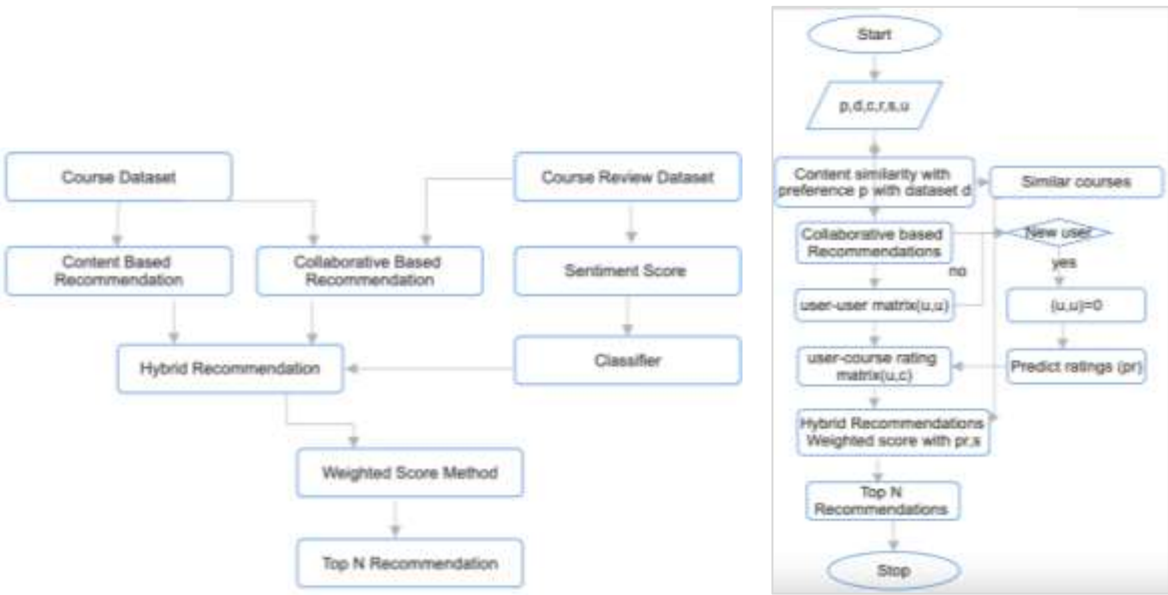
Here, TF (t, d) represents the TF of term t in document d, while f (t, d) denotes the frequency of term t in document d. The term |d| signifies the total number of terms in document d. IDF assesses the occurrence of unique terms across a collection of documents. It is calculated as in (2).

$$IDF(t,d) = log(\frac{N}{n(t,D)}) \tag{2}$$

Where IDF (t, D) is the IDF of term t in the document collection D. N represents the total number of documents in the collection D, and n (t, D) is the number of documents in the collection D that contain the term t. The TF-IDF score for a term t within d is computed as the product of TF and IDF [28] as in (3).

$$TF - IDF(t,d,D) = TF(t,d)\ X\ IDF(t,D) \tag{3}$$

TF-IDF score provides a numerical representation of the importance of a term within a specific document, accounting for the term's frequency within that document and its uniqueness across the entire document collection [29]. Each query from the chatbot after pre-processing is represented as a TF-IDF vector, where each term in the document corresponds to a component of the vector. The user query is represented by a TF-IDF vector, and to find the similarity between the user query and the documents, cosine similarity is used between the user's TF-IDF vector and the TF-IDF vectors of the documents. This identifies the user's preferences, matching the content of the related documents. The documents with higher cosine similarity values are considered more like the user's preferences and are recommended as they are likely to match the user's interests. TF-IDF represents the similarity between documents and user profiles using vector representations. This approach can rank recommendations for courses on a MOOC platform based on user preferences and content similarity.



(a) (b)

Figure 4. Hybrid recommendation (a) working model and (b) algorithmic flow of information

Recommendations based on user-user interactions are a form of collaborative filtering widely employed in recommendation systems. It relies on the idea that users with similar behaviour or preferences in the past are likely to continue to have similar interests in the future. These methods make recommendations by considering the actions and preferences of users and comparing them to other users in the system [30]. Learners often refer to course reviews to find positive feedback or higher ratings from fellow learners as an indicator of the quality and desirability of courses. Conversely, negative comments or lower ratings gauge whether a course should be avoided. Consequently, both numerical ratings and textual reviews can be leveraged to improve course recommendations, offering a more comprehensive approach beyond solely relying on numerical ratings. One widely used method in collaborative filtering recommendation systems is matrix factorization using latent factor models [25]. It decomposes a user-course matrix into two matrixes: the user-user matrix and the user-course ratings matrix. If a course is found interesting by a user, it will be recommended to users who have shared similar interests in the past [10]. To achieve this, courses and users' factors are represented in the vectors of the user-user matrix and the ratings of the user-course matrix. A high value in the matrix between course and user factors leads to a recommendation.

The user matrix captures similarity between peer learners. When a new user is introduced, it calculates its similarity to the current users and adjusts the relevant matrix items accordingly. The challenge arises from the fact that users tend to rate courses differently, some users may assign high ratings to a course while others might rate it lower. To address this variation, the mechanism improved version of cosine similarity is the adjusted cosine similarity, it subtracts the average ratings given by each user from their individual ratings for various courses. The correlation threshold technique forms a cluster keeps only the similar users who exceeds the threshold. Every cell in the matrix represents the similarity between a cluster member and another cluster member [27]. The measure of similarity is established through cosine similarity, a metric that evaluates similarity by considering the angle between two vectors. The computation of this cosine-based similarity measure as in (4).

$$sim(v1, v2) = \frac{\sum_{a=1}^{m'} w(kv_1a) * w(kv_2a)}{\sqrt{\sum_{a=1}^{m'} (w(kv_1a))^2 * \sum_{a=1}^{m'} (w(kv_2a))^2}} \tag{4}$$

Table 1. User-user similarity matrix

	v_1	v_2	v_3	…	v_N
v_1	-	sim $_{v1, v2}$	sim $_{v1, v3}$	…	sim $_{v1, vN}$
v_2	-	-	sim $_{v2, v3}$	…	sim $_{v2, vN}$
v_3	-	-		…	sim $_{v3, vN}$
	…	…	…	…	…
v_n	-	-	-	…	-

where m' represents the union of keywords v_1, $v_2$and $w(kv_1a)$ and $w(kv_2a)$ denotes the weight of the keyword of m' in v_1, v_2. sim (vi, vj) relates to the resemblance among cluster members vi and vj to identify other members that surpasses a predetermined threshold value. Following the matrix update, the cluster leader then consolidates profiles for every member within the cluster [27]. For each cluster member, the leader reviews the user-user similarity matrix, as depicted in Table 1. A user-course rating matrix as given in the Table 2 containing the course ratings with user u= {u_1, u_2, u_3, … u_n} given the course ratings for the courses C= {c_1, c_2, c_3, ... c_n} rating R as represented as {$r^R u_{1,\ C1}$, $r^R u_{2,\ C2}$, … $r^R u_{n,\ Cn}$}. To forecast the ratings a user might assign to each course ("C") not yet assessed, calculate a weighted average of the ratings given to "C" by users exhibiting similarity received from user-user matrix. This involves utilizing the user-course similarity matrix formed as in (5).

Table 2. user-course rating matrix

	${c_1}^R$	${c_2}^R$	${c_3}^R$	…	${c_n}^R$
u_1	$r^R u_{1,\ C1}$	$r^R u_{1,\ C2}$	$r^R u_{1,\ C3}$		$r^R u_{1,\ Cn}$
u_2	$r^R u_{2,\ C1}$	$r^R u_{2,\ C2}$	$r^R u_{2,\ C3}$		$r^R u_{2,\ Cn}$
u_3	$r^R u_{3,\ C1}$	$r^R u_{3,\ C2}$	$r^R u_{3,\ C3}$		$r^R u_{3,\ Cn}$
	…	…	…	…	…
u_n	$r^R u_{n,\ C1}$	$r^R u_{n,\ C2}$	$r^R u_{n,\ C3}$		$r^R u_{n,\ Cn}$

$$r^R u_{target,\ Cn}{}^R = \frac{\sum_{n=1}^{n} (r_{un,c_n}{}^R) * sim(u_{target,un})}{Count(sim(u_{target,un}))} \tag{5}$$

User-course rating matrix is used to recommend courses to new users based on ratings given by similar users overcomes cold start issues of collaborative filtering.

2.5. Sentiment analysis

Sentiment analysis categories textual reviews of users and assigns a sentiment score. Elahi *et al.* [23] applied BERT sentiment analysis to the hybrid product recommendation system and evaluated it using Amazon games and video datasets. To achieve the research objective's, VADER is employed to analyse the sentiments conveyed in the text reviews of users. Extracting the sentiment on identified topics from elements influencing user satisfaction, the comments are categorized as positive, negative, and neutral [30]. Random forest (RF) and logistic regression (LR) classifiers are used for analysing sentiments from the available user reviews. The RF classification method results in the best accuracy in comparison to the LR method. The course review dataset facilitates the retrieval of more than one million reviews by searching for specific courses. The proposed approach uses both ratings and sentiments from reviews in a hybrid recommender system to improve accuracy.

2.6. Top N recommendations

The hybrid course recommendation using sentiment analysis in the proposed Algorithm 1 encompasses the subsequent steps. Cosine similarity is used for content-based recommendation in step 1 and 2. The recommended system constructs a user-user similarity matrix and a user-course matrix containing the course ratings for each user is captured using steps 3 to 5. Subsequently, the adjusted cosine similarity algorithm is applied to calculate the similarity among the users in step 6. The ranking score for a course recommendation calculated with the predicted rating (PR) and sentiment score(s) is normalized on a scale of 1–5 is used in step 7 and 8. These values act as the input for the hybrid recommendation algorithm.

Algorithm 1. Hybrid course recommendation using sentiment analysis

```
Input: User u Preferences from chatbot p, web scraped dataset d with courses c, rating r
Output: Top N Recommendations(ui, ci, ri)
Step.1: For i <- 1 to c do
Step.2: calculate sim(ci,p) using TF-IDF outputs Top recommendations
Step.3: if r(ui,ci) is available go to step (5) else goto step(4)
Step.4: if r(ui,ci)=nan replace with r(ui, ci)=0 goto step (6)
Step.5: for i in 1 to u do
            for j in 1 to u do
               Creating the user similarity matrix using the correlation threshold (ui, uj)
Step.6: for i in 1 to u do
            for j in 1 to c do
               Creating the user course rating matrix using function (ui, cj)
                              Predict Rating (user-course) using adjusted cosine similarity
(pr)
Step.7: if new user not rated any course weighted average of the ratings given to C by peer
user with (5).
Step.8: Get Sentiment score from the similar user(s)
Step.9: Assign weights w1, w2 for predicted rating(pr) and sentiment(s)
Step 10: Find top n recommendations using with W1 * (pr)+ W2 *s. where W2 is twice as W1.
```

The weighted score method can be quite effective in building hybrid recommendations where multiple criteria are to be considered. It determines the relative importance of each criterion by assigning a numerical weight. Typically, users place greater emphasis on reviews than on ratings. To reflect this preference, W1 and W2 are assigned, and the criteria reviews are given double the weight of ratings in the ranking score calculations of Top N recommendations.

3. RESULTS AND DISCUSSION

This study shows the use of hybrid strategies for personalized course recommendation, while earlier studies suffered from a cold start issue and did not explicitly capture individual user preferences. The proposed hybrid recommendation algorithm was tested on a web scraped course dataset with 13K courses and a review dataset with 100K values. The evaluation results acquired in the hybrid recommendation framework are presented after thorough investigations. By comparing the results with those presented in [8], [20], and [23], employing the metrics of accuracy, recall, F1-value, and RMSE. It is implemented using the Python language embedded in the TensorFlow package. The following sections explain the results obtained by the methods used in the framework.

3.1. Dataset

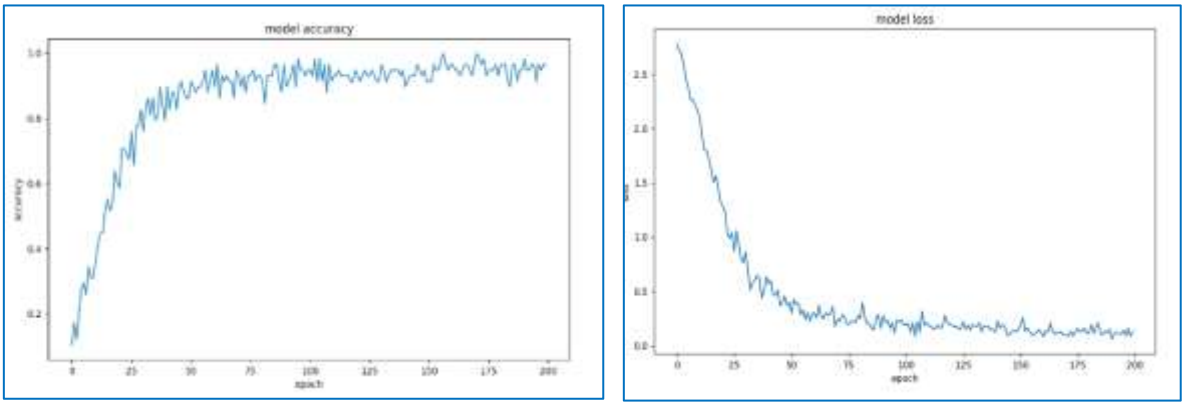
The web scraping technique with the Selenium tool is used to obtain data from MOOC websites for recommendations. Many of the course recommendation systems in the literature [17], [18] use MOOC data, which is not updated often. To work with real-time data, course name, university, course description, course URL, level, duration, rating, specialization, language, subtitles, partner type, domain information, expected completion time, description, specialization URL, skills learned, language, and sub-title are web scraped and stored in the dataset. Course name, university, description, course URL, level, duration, rating, specialization, and language columns are considered for data analysis, and other data is ignored because of its lower significance and higher number of null values. A review dataset with fields course name, course URL, reviewer name, and reviews is obtained from Kaggle [13]. The attributes of the dataset are given in Table 3.

Table 3. Attributes of the dataset

Fields	Description
Course name	Name of the course
University	University offering the course
Description	Overview of topics and content of course
Course URL	Website link to access the course
Level	Course difficulty-beginner, intermediate, advanced
Duration	Time required for course completion
Rating	Average rating given by all the users of the course
Specialization	Focused area of with more courses on one domain
Language	Medium of communication
Reviewer name	Anonymous reviewer name
Review content	Textual description of learner’s feedback on course

3.2. Chatbot

Chatbots are used to identify the personalised learning requirements through interaction with learners. A two-stage Bayesian algorithm to recommend customized learning materials according to learners’ requirements was developed by Yao and Wu [31] with accuracy rate of 90%. The proposed deep learning chatbot provides good responses on the trained intent and responses with 75% of the data for training and 25% testing process. Figure 5. depicts the accuracy and loss value of chatbot model. The deep learning model aligns with the user’s choice and is executed for 200 epochs, resulting in an accuracy of 96.5%, as shown in Figure 5(a). The loss values shown in Figure 5(b) decrease after 50 iterations.

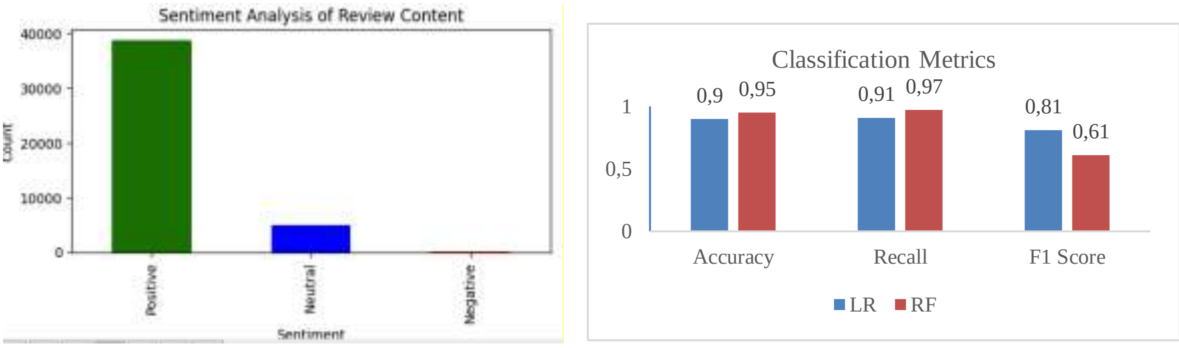


(a) (b)

Figure 5. Chatbot (a) accuracy and (b) loss value

3.3. Sentiment analysis

Course reviews were analysed using the VADER model and its data exploration given in Figure 6. It indicates that 66.8% of the sentiments were positive and 33.2% were negative and neutral as shown in Figure 6(a). The input X features from the document are fed into the various classification models such as LR and RF algorithm with 75% of the data for training and the remaining 25% for testing. Random Forest algorithms are considered the best to estimate the probability of a positive, neutral, or negative sentiment, evaluate the odds, and calculate the probability of success divided by the probability of failure [32]. Figure 6(b) illustrates the visualization of the accuracy, precision, recall, and F1-score of the RF, which performs comparatively better in terms of accuracy and recall value.



(a) (b)

Figure 6. Sentiment analysis (a) distribution of sentiments and (b) accuracy of classification

3.4. Hybrid algorithm for top N recommendations

Results and values from the web scraping dataset, chatbot input, and sentiment score are applied to the hybrid recommendation algorithm and iterated n times. Using the weighted score method, weights W1 and W2 are assigned, and the top N recommendations computed with N value is equal to pr*W1+s*W2. The predicted rating and sentiment score yield better recommendations compared to the existing methods. The highest-value N-value course taken for Top N recommendations. Based on the results is given in Figure 7, the hybrid recommendation framework recommends Top N courses with and without sentiment analysis.

Figure 7. Hybrid recommendation algorithm RMSE value

The comparison between user-based recommendations and course-based recommendations reveals user-based recommendations with sentiment analysis performs better with RMSE value of 0.541 while comparing with course-based recommendations. However, in depth analysis to be carried out in optimising the methods used in the framework. Further studies may explore integrating the proposed framework with large language models may enhance the accuracy of recommendations. The findings reveal conclusive evidence for the use of a hybrid framework for real-time course recommendations across multiple MOOC platforms.

4. CONCLUSION

This paper studies a novel hybrid framework of recommendation that integrates deep learning, sentiment analysis, and user-based recommendations. This approach to MOOCs can offer course suggestions depending on the requirements of the user. The proposed method combines user-based recommendation with sentiment analysis to solve the problem of existing recommendation methods. The proposed method focuses on the web scraping of data's, the pre-processing of datasets, and the development of deep learning chatbots. This recommended model is designed to use the advantages of sentiment analysis in processing text reviews rather than relying on user ratings. Finally, user-based recommendations with weighted score methods are developed to improve the performance of recommendations. The experimental results have demonstrated that the proposed framework performs better than the existing methods with a reduced RMSE value. However, this paper also has some limitations. This paper lacks the deployment of large language models in implementation, recommendation system must be tested on any education platform for providing personalized learning experience. In the future, we will address the limitations mentioned.

REFERENCES

[1] H. Zhang, X. Shen, B. Yi, W. Wang, and Y. Feng, "KGAN: knowledge grouping aggregation network for course recommendation in MOOCs," *Expert Systems with Applications*, vol. 211, p. 118344, Jan. 2023, doi: 10.1016/j.eswa.2022.118344.

[2] A. Khalid, K. Lundqvist, and A. Yates, "A literature review of implemented recommendation techniques used in Massive Open online Courses," *Expert Systems with Applications*, vol. 187, p. 115926, Jan. 2022, doi: 10.1016/j.eswa.2021.115926.

[3] N. Vedavathi and K. M. A. Kumar, "E-learning course recommendation based on sentiment analysis using hybrid Elman similarity," *Knowledge-Based Systems*, vol. 259, p. 110086, Jan. 2023, doi: 10.1016/j.knosys.2022.110086.

[4] R. Campos, R. P. dos Santos, and J. Oliveira, "Providing recommendations for communities of learners in MOOCs ecosystems," *Expert Systems with Applications*, vol. 205, p. 117510, Nov. 2022, doi: 10.1016/j.eswa.2022.117510.

[5] K. Han, "Personalized news recommendation and simulation based on improved collaborative filtering algorithm," *Complexity*, vol. 2020, pp. 1–12, Oct. 2020, doi: 10.1155/2020/8834908.

[6] A. A. N. Avval and A. Harounabadi, "A hybrid recommender system using topic modeling and prefixspan algorithm in social media," *Complex and Intelligent Systems*, vol. 9, no. 4, pp. 4457–4482, Aug. 2023, doi: 10.1007/s40747-022-00958-5.

[7] R.Alatrash, R. Priyadarshini, H. Ezaldeen, and A. Alhinnawi, "Augmented language model with deep learning adaptation on sentiment analysis for E-learning recommendation," *Cognitive Systems Research*, vol. 75, 2022, doi: 10.1016/j.cogsys.2022.07.002.

[8] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simulation Modelling Practice and Theory*, vol. 113, p. 102375, Dec. 2021, doi: 10.1016/j.simpat.2021.102375.

[9] C. De Medio, C. Limongelli, F. Sciarrone, and M. Temperini, "MoodleREC: a recommendation system for creating courses using the moodle e-learning platform," *Computers in Human Behavior*, vol. 104, p. 106168, Mar. 2020, doi: 10.1016/j.chb.2019.106168.

[10] V. M. Romero, B. D. Santiago, and J. M. Z. Nuevo, "A hybrid recommendation scheme for delay-tolerant networks: The case of digital marketplaces," *Array*, vol. 19, p. 100299, Sep. 2023, doi: 10.1016/j.array.2023.100299.

[11] N. Pavitha *et al.*, "Movie recommendation and sentiment analysis using machine learning," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 279–284, Jun. 2022, doi: 10.1016/j.gltp.2022.03.012.

[12] B. Wu and L. Liu, "Personalized hybrid recommendation algorithm for MOOCs based on learners' dynamic preferences and multidimensional capabilities," *Applied Sciences (Switzerland)*, vol. 13, no. 9, p. 5548, Apr. 2023, doi: 10.3390/app13095548.

[13] T. UŽDAVINYS, "Coursera specialization dataset 2023-FEB," *kaggle*, 2023. https://www.kaggle.com/datasets/uzdavinys/coursera-specialization-dataset-2023-feb.

[14] H. Ezaldeen, R. Misra, S. K. Bisoy, R. Alatrash, and R. Priyadarshini, "A hybrid E-learning recommendation integrating adaptive profiling and sentiment analysis (double)," *Journal of Web Semantics*, vol. 72, p. 100700, Apr. 2022, doi: 10.1016/j.websem.2021.100700.

[15] S. Ahmadi, Z. Nourmohamadzadeh, and B. Amiri, "A hybrid DEMATEL and social network analysis model to identify factors affecting learners' satisfaction with MOOCs," *Heliyon*, vol. 9, no. 7, p. e17894, Jul. 2023, doi: 10.1016/j.heliyon.2023.e17894.

[16] S. Amin, M. I. Uddin, A. A. Alarood, W. K. Mashwani, A. Alzahrani, and A. O. Alzahrani, "Smart E-learning framework for personalized adaptive learning and sequential path recommendations using reinforcement learning," *IEEE Access*, vol. 11, pp. 89769–89790, 2023, doi: 10.1109/ACCESS.2023.3305584.

[17] B. Li *et al.*, "A personalized recommendation framework based on MOOC system integrating deep learning and big data," *Computers and Electrical Engineering*, vol. 106, p. 108571, Mar. 2023, doi: 10.1016/j.compeleceng.2022.108571.

[18] N. González-Castro, P. J. Muñoz-Merino, C. Alario-Hoyos, and C. D. Kloos, "Adaptive learning module for a conversational agent to support MOOC learners," *Australasian Journal of Educational Technology*, vol. 37, no. 2, pp. 24–44, May 2021, doi: 10.14742/AJET.6646.

[19] M. P. C. Lin and D. Chang, "CHAT-ACTS: a pedagogical framework for personalized chatbot to enhance active learning and self-regulated learning," *Computers and Education: Artificial Intelligence*, vol. 5, p. 100167, 2023, doi: 10.1016/j.caeai.2023.100167.

[20] J. W. Tzeng, N. F. Huang, A. C. Chuang, T. W. Huang, and H. Y. Chang, "Massive open online course recommendation system based on a reinforcement learning algorithm," *Neural Computing and Applications*, Jun. 2023, doi: 10.1007/s00521-023-08686-8.

[21] U. Sujatha and R. Gunasundari, "A hybrid personalized MOOC course recommendation model," *European Chemical Bulletin*, 2023, doi: 10.31838/ecb/2023.12.s1-B.382.

[22] S. Kumar and U. B. Roy, "A technique of data collection: web scraping with python," in *Statistical Modeling in Machine Learning: Concepts and Applications*, Elsevier, 2022, pp. 23–36.

[23] M. Elahi, D. K. Kholgh, M. S. Kiarostami, M. Oussalah, and S. Saghari, "Hybrid recommendation by incorporating the sentiment of product reviews," *Information Sciences*, vol. 625, pp. 738–756, May 2023, doi: 10.1016/j.ins.2023.01.051.

[24] Y. Lin *et al.*, "Hierarchical reinforcement learning with dynamic recurrent mechanism for course recommendation," *Knowledge-Based Systems*, vol. 244, p. 108546, May 2022, doi: 10.1016/j.knosys.2022.108546.

[25] M. V. Farashah, A. Etebarian, R. Azmi, and R. E. Dastjerdi, "A hybrid recommender system based-on link prediction for movie baskets analysis," *Journal of Big Data*, vol. 8, no. 1, p. 32, Dec. 2021, doi: 10.1186/s40537-021-00422-0.

[26] R. K. Rajasekaran and A. I. Iliev, "Hybrid recommendation system," in *Lecture Notes in Networks and Systems*, vol. 651 LNNS, 2023, pp. 1–11.

[27] S. Yin, K. Yang, and H. Wang, "A MOOC courses recommendation system based on learning behaviours," in *ACM International Conference Proceeding Series*, May 2020, pp. 133–137, doi: 10.1145/3393527.3393550.

[28] F. Colace, M. De Santo, L. Greco, V. Moscato, and A. Picariello, "A collaborative user-centered framework for recommending items in Online Social Networks," *Computers in Human Behavior*, vol. 51, pp. 694–704, Oct. 2015, doi: 10.1016/j.chb.2014.12.011.

[29] R. J. Kuo and S. S. Li, "Applying particle swarm optimization algorithm-based collaborative filtering recommender system considering rating and review," *Applied Soft Computing*, vol. 135, p. 110038, Mar. 2023, doi: 10.1016/j.asoc.2023.110038.

[30] A. Tanwar and D. K. Vishwakarma, "A deep neural network-based hybrid recommender system with user-user networks," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15613–15633, Apr. 2023, doi: 10.1007/s11042-022-13936-3.

[31] C.-B. Yao and Y.-L. Wu, "Intelligent and interactive Chatbot based on the recommendation mechanism to reach personalized learning," *International Journal of Information and Communication Technology Education*, vol. 18, no. 1, pp. 1–23, Dec. 2022, doi: 10.4018/ijicte.315596.

[32] S. Munuswamy, M. S. Saranya, S. Ganapathy, S. Muthurajkumar, and A. Kannan, "Sentiment analysis techniques for social media-based recommendation systems," *National Academy Science Letters*, vol. 44, no. 3, pp. 281–287, Jun. 2021, doi: 10.1007/s40009-020-01007-w.

BIOGRAPHIES OF AUTHORS

Sujatha Uthamaraj is a Research scholar in the Department of Computer Science at Karpgagam Academy of Higher Education, Coimbatore. She holds Mphil from Bharathiyar Univeristy, Coimbatore and Master's in Computer Applications from Karpgagam Academy of Higher Education, Coimbatore. Having 16 years' experience of in teaching, her research interest includes learning analytics, machine learning, and recommendation Systems. She is member of ISTE. She has authored more than 7 publications. She can be contacted at email: sujathau@gmail.com.

Dr. Gunasundari Ranganathan is presently working as a Professor in the Department of Computer Applications, Karpagam Academy of Higher Education, Coimbatore, her research interest includes Data mining and Learning Analytics. She has authored or coauthored more than 90 publications: with 7 H-index and more than 100 citations. Her research interests, machine learning, and data mining. She can be contacted at email: gunasoundar04@gmail.com.