Online Book Recommendation System

Nipuni Jayathilake
Dept. Computer Science & Engineering
University of Moratuwa
Sri Lanka
nipuni.21@cse.mrt.ac.lk

Kaushal Kahapola
Dept. Computer Science & Engineering
University of Moratuwa
Sri Lanka
kaushal.21@cse.mrt.ac.lk

Lakindu Kariyawasam
Dept. Computer Science & Engineering
University of Moratuwa
Sri Lanka
lakindu.21@cse.mrt.ac.lk

***Keywords*—book recommendation, collaborative filtering, lightgcn**

I. Introduction

With the growing number of books, finding relevant content is challenging. We developed an **Online Book Recommendation System** using a **hybrid approach** combining **content-based** and **collaborative filtering**, ensuring **personalized and diverse** recommendations.

II. Literature Review

Previous work on book recommendations includes content-based filtering using textual attributes and collaborative filtering based on user interactions. Traditional keyword-based systems struggle to fully capture user preferences.

Hybrid approaches, which combine content-based and collaborative filtering, have shown promise in addressing this issue by providing a balance between diversity and personalization in recommendations. Recent works have also used graph-based recommendation models, such as LightGCN, which have gained attention for their effectiveness in modeling user-item interactions. Additionally, the use of embeddings, such as text-embedding-004, has proven beneficial in generating high-quality content representations for efficient similarity searches.

Graph-based models such as LightGCN have emerged as a powerful method in collaborative filtering, improving traditional techniques by using graph convolutions to learn latent representations of users and items. However, integrating these approaches into a comprehensive hybrid model and assessing their performance in real-world scenarios remains a key challenge.

Recent advances in hybrid recommendation systems further suggest that incorporating deeper semantic analysis through advanced embeddings or incorporating additional data types (e.g., social media activity) could offer substantial improvements in system accuracy and personalization.

III. Methods and Materials

A. Datasets and Preprocessing

The final model was trained on the Amazon Book Dataset, composed of two tables:

- **Books Data:-** *title, description, authors, image, preview link, publisher, categories, ratingsCount*
- **Book Ratings:-** *id, title, author, price, userId, profileName, review/score, review/summary*

Preprocessing steps included:

- **Missing Values -** Filled missing entries.
- **Normalization -** Lowercased titles and descriptions; removed unwanted characters and extra spaces.
- **Inconsistencies -** Resolved using `rapidfuzz` for fuzzy matching; similar titles replaced and duplicates removed.
- **Filtering -**
  - Removed books with no reviews or ratingsCount ¡ 10.
  - Retained only books with review/score ≥ 4.
  - Removed users who rated fewer than 5 books.

Our system combines content-based and collaborative filtering.

B. Content-Based Filtering

Content-based filtering recommends books similar to those the user has shown interest in, based on textual features like genre, author, and description. We use the **text-embedding-004** model to generate these vectors. The similarity between books i and j is computed using cosine similarity:

$$\text{cosine\_similarity}(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}$$

Books with the highest similarity scores are recommended, ensuring relevance of the content.

C. Collaborative Filtering: LightGCN

Collaborative filtering recommends books based on user preferences. LightGCN improves traditional methods by using graph convolutions to model user-item relationships.

Let's define the user-item interaction graph G, where: - U represents users, - I represents books (items), - $E \subseteq U \times I$ represents the edges (user-item interactions).

Each user $u \in U$ and book $i \in I$ is represented as a latent vector $\mathbf{u}$ and $\mathbf{i}$ respectively. The goal of LightGCN is to learn these latent vectors by performing graph convolutions on the

https://doi.org/10.31705/ADScAI.2025.46

TABLE I
PERFORMANCE METRICS

Precision@5	Recall@5	NDCG@5	ROC AUC
0.0987	0.1318	0.2964	0.7747

user-item graph. The LightGCN model refines the embeddings using the following operation:

$$\mathbf{u}^{(k+1)} = \mathbf{U} + \sum_{i \in \mathcal{N}(u)} \mathbf{i}^{(k)} \quad (1)$$

Where: - $\mathbf{u}^{(k+1)}$ is the updated representation of user u after the $(k+1)$-th convolutional layer. - $\mathbf{U}$ is the initial user embedding. - $\mathcal{N}(u)$ is the set of books with which the user u has interacted. - $\mathbf{i}^{(k)}$ is the embedding of book i after the k-th convolutional layer.

Similarly, the item embedding is updated in a parallel fashion:

$$\mathbf{i}^{(k+1)} = \mathbf{I} + \sum_{u \in \mathcal{N}(i)} \mathbf{u}^{(k)} \quad (2)$$

Where: - $\mathbf{i}^{(k+1)}$ is the updated representation of item i.

After several layers of graph convolution, the user and item embeddings are used to calculate the predicted rating or similarity score between user u and item i:

$$\hat{r}_{ui} = \mathbf{u}^T \mathbf{i} \quad (3)$$

Where: - $\hat{r}_{ui}$ is the predicted rating for user u and item i. - $\mathbf{u}$ and $\mathbf{i}$ are the learned latent vectors for user u and book i.

The model is trained to minimize the error between predicted ratings and actual user ratings using a loss function such as Mean Squared Error (MSE).

D. Hybrid Approach

The final recommendation score combines content-based and collaborative filtering (LightGCN) outputs.

$$S_{ui} = \alpha \cdot S_{CB} + (1 - \alpha) \cdot S_{CF} \quad (4)$$

Where: - S_{ui} is the final recommendation score for user u and item i. - S_{CB} is the score from content-based filtering. - S_{CF} is the score from collaborative filtering. - α is a hyperparameter controlling the balance between the two approaches.

The books with the highest final scores are recommended to the user.

IV. RESULTS AND DISCUSSION

The system was evaluated using standard recommendation metrics.

A. Performance Metrics

The following metrics were used to evaluate recommendation quality:

These results indicate that the hybrid approach significantly improves the accuracy of the recommendation compared to traditional methods.

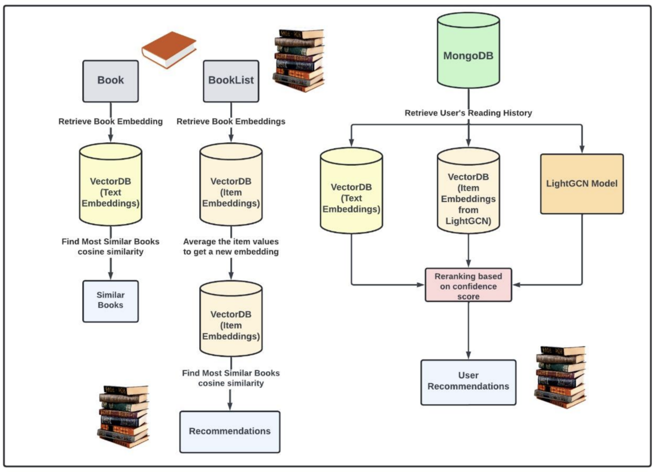

Fig. 1. Recommendations are provided to users using different approaches

B. System Implementation

The system architecture includes a backend for authentication, data handling, and recommendation logic; a frontend for user interaction; a database for storing user data, book metadata, and logs; and a vector store for managing embeddings and enabling similarity-based recommendations.

V. CONCLUSION

We developed a hybrid recommendation system combining content-based filtering and LightGCN, offering personalized recommendations. The system shows promise for real-world applications, with strong performance based on several evaluation metrics.

Several limitations exist. Content-based filtering may favor popular books, while collaborative filtering can amplify existing user biases. The system may struggle with scalability as the dataset grows, affecting training time and performance. Cold start issues for new users or books can lead to less accurate early recommendations. Balancing novelty and relevance is challenging too much personalization may reduce diversity. Additionally, sparse user-item interactions can impact collaborative filtering accuracy.

ACKNOWLEDGMENT

This research was carried out as part of the Department of Computer Science and Engineering of the University of Moratuwa. We thank our mentors and peers for their valuable feedback and insights.

REFERENCES

[1] G. Karypis, "Evaluation of Item-Based Top-N Recommendation Algorithms," SIAM Data Mining, 2001.
[2] X. He, K. Deng, X. Wang, Y. Li, and T. Chua, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in Proc. ACM SIGIR, 2020.
[3] A. Said and A. Bellogin, "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks," in RecSys, 2014.