Neurocomputing 649 (2025) 130753

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Recommendation systems with LLM-based semantic embeddings and FAISS similarity search

Seema Safar [a,*], Babita Roslind Jose [a], Jimson Mathew [b], T. Santhanakrishnan [c]

[a] School of Engineering, Cochin University of Science and Technology, Kochi, 682022, Kerala, India
[b] Department of Computer Science and Engineering, Indian Institute of Technology, Patna, Patna, 801106, Bihar, India
[c] Naval Physical and Oceanographic Laboratory, Defense Research and Development Organisation, Kochi, Kerala, India

ARTICLE INFO

Communicated by Y. Long

Keywords:
Information retrieval
LLM
Semantic embeddings

ABSTRACT

Content-based recommendation systems have gained significant attention for their ability to provide personalized suggestions by analyzing item descriptions. Leveraging the power of large language models (LLMs), this research introduces a novel recommendation approach that generates high-quality semantic embeddings to facilitate efficient similarity-based retrieval for Top-N recommendations. The proposed method capitalizes on the deep contextual understanding of LLMs to capture intricate semantic relationships within item content, thereby enhancing recommendation relevance. Furthermore, the system integrates FAISS (Facebook AI Similarity Search) to optimize similarity search, enabling faster and more scalable retrieval of relevant recommendations. To evaluate its effectiveness, the system is tested on four diverse real-world datasets: Yelp, Amazon Beauty, MovieLens, and LastFM, covering multiple domains. Performance is assessed using widely adopted evaluation metrics, including Normalized Discounted Cumulative Gain (NDCG), Precision, Recall, Hit Rate (HR), F1-Score and business-relevant evaluation measures. Extensive experimental results demonstrate that the proposed method, augmented with FAISS, consistently outperforms the existing state-of-the-art recommendation techniques. The code supporting this code is publicly available at: https://github.com/seemasafar/Reco-System-Using-LLM

1. Introduction

Recommendation systems have become a cornerstone of numerous industries, providing personalized experiences by predicting items users may like based on historical data. From e-commerce platforms suggesting products to music services curating playlists, these systems have fundamentally transformed digital interactions. Traditional recommendation approaches, primarily based on collaborative filtering, content-based filtering, or hybrid models, have been effective in many scenarios. However, they often suffer from challenges such as data sparsity, an inability to incorporate contextual information, and scalability limitations. As a result, there is growing interest in leveraging deep learning techniques, particularly large language models (LLMs), to enhance recommendation quality.

Recent advancements in LLMs have demonstrated their ability to capture intricate patterns and semantic relationships across diverse natural language processing (NLP) tasks. Trained on vast amounts of textual data, these models generate rich, context-aware embeddings that more accurately represent user preferences and item characteristics. Unlike traditional matrix factorization and heuristic-based methods, deep learning-based embeddings offer the potential to enhance personalization by capturing higher-order semantic information. However, while LLMs have been extensively applied in NLP and AI-driven applications, their role in recommendation systems remains relatively underexplored. Key challenges include computational complexity, the integration of textual embeddings into structured recommendation pipelines, and the lack of benchmark comparisons between different LLM-based embedding generation models.

This study aims to address this gap by systematically investigating the effectiveness of three state-of-the-art LLMs – LLaMA2, Mistral, and Phi-3 mini – in generating embeddings for recommendation systems. These models were chosen for their ability to encode deep semantic information from textual data, which is crucial for improving recommendation accuracy. By leveraging these embeddings, we propose a novel recommendation framework that integrates LLM-based representations with modern recommendation algorithms, particularly

* Corresponding author.
E-mail addresses: seemasafar@cusat.ac.in (S. Safar), babitajose@cusat.ac.in (B.R. Jose), jimson@iitp.ac.in (J. Mathew), tsanthan.npol@gov.in (T. Santhanakrishnan).

https://doi.org/10.1016/j.neucom.2025.130753
Received 27 February 2025; Received in revised form 27 May 2025; Accepted 7 June 2025
Available online 24 June 2025

in a sequential recommendation context. This approach enables dynamic adaptation to users' evolving preferences, moving beyond static recommendation strategies.

In a nutshell, the key contributions of the proposed work may be summarized as follows:

- The proposed work systematically constructs a structured textual representation for each item in the dataset. This representation integrates essential item attributes, including item ID, item category, item description, and URL. By standardizing the textual data, the model ensures that embeddings capture meaningful semantic relationships, thereby enhancing the quality of similarity-based retrieval.
- For automated generation and efficient handling of embeddings, the implementation seamlessly integrates an external embedding generation service via API calls to LLM, in order to obtain semantic vector representations of item descriptions. The system is optimized to minimize redundant computation by checking for pre-existing embeddings stored in a NumPy file. If a mismatch between dataset size and stored embeddings is detected, embeddings are regenerated to ensure data integrity. Additionally, the system incorporates robust exception handling to manage network failures or API request issues, preventing disruptions in the embedding generation process.
- In order to ensure a high performance similarity search, the system employs FAISS (Facebook AI Similarity Search) to efficiently perform nearest-neighbor search over high-dimensional embedding vectors from LLM. The implementation supports both the loading of pre-trained FAISS indices and the dynamic creation of a new index when necessary. By leveraging FAISS's L2 distance metric for similarity measurement, the system ensures rapid and scalable retrieval of semantically similar products, significantly improving recommendation efficiency compared to brute-force similarity computations.
- For dynamic product selection for personalized recommendations, the recommendation process is designed to provide a dynamic and unbiased evaluation by randomly selecting an item from the dataset as the query item. This randomized approach allows for broad coverage of the dataset, ensuring that recommendations are not biased toward any particular item subset. It enhances the evaluation process by simulating real-world recommendation scenarios where different users search for different items.

To assess the effectiveness of the proposed model, the performance is evaluated against seventeen state-of-the-art baselines across four standard diverse datasets: Yelp, Amazon Beauty, MovieLens (ML-1M), and LastFM. Experimental results consistently demonstrate the superiority of our approach, as the proposed model achieves significant improvements across multiple evaluation metrics. Specifically, on the Yelp dataset, our LLM-enhanced recommendation model outperforms both traditional and graph-based collaborative filtering methods, achieving substantial performance gains.

2. Related works

Recent advancements in Large Language Models (LLMs) have significantly transformed recommendation systems by enhancing user-item interactions with deep semantic understanding. Traditional recommendation approaches, such as collaborative filtering and content-based filtering, often struggle with cold-start problems, data sparsity, and scalability. Some innovations in traditional approaches include a lightweight recommendation framework leveraging Graph Neural Networks (GNNs) [1] to model user-item interactions with reduced computational complexity, making it suitable for real-time applications. Another study on aspect-based sentiment analysis [2], enhance the accuracy of recommendations by capturing fine-grained user opinions from reviews. Some approaches [3] include integrating social relationship data and item interaction patterns, resulting in a more holistic and socially informed recommendation model.

In contrast, LLM-based models leverage pre-trained contextual embeddings, multimodal data, and prompt-based learning to generate more personalized and context-aware recommendations. Advancements in network design, semantic analysis, and the integration of large language models (LLMs) have significantly influenced recommendation systems and related domains. Duan et al. [4] introduced Dynamic Unary Convolution within Transformer architectures, enhancing the adaptability and efficiency of attention mechanisms by dynamically adjusting convolution kernels based on input. This work highlights the evolving landscape of efficient network designs that are highly relevant to semantic-based modeling, as seen in LLM-driven applications.

Similarly,Duan et al. [5] proposed a novel visual-language framework approach that demonstrates the effectiveness of multi-modal fusion and semantic analysis, principles that are increasingly important in systems combining text (e.g., text descriptions) and structured data (e.g., genres) for recommendation tasks. Additionally, Chu et al. [6]introduced Union-Domain Knowledge Distillation to enhance underwater acoustic target recognition by transferring knowledge across domains. Another notable research work include Hu et al. [7], explores enhanced ADHD detection by embedding frequency information into a visual-language framework, highlighting the potential of multimodal learning for specialized classification tasks. This technique underlines the importance of cross-domain semantic alignment and distilled representation learning, both of which are critical for improving generalization in hybrid recommendation systems and semantic retrieval tasks. Recent research has highlighted the transformative potential of Large Language Models (LLMs) in guiding multi-modal image fusion by enabling high-level semantic understanding.

In particular, LLMs integrated with generative vision-language models have demonstrated their ability to align visual content with textual guidance, leading to more coherent and semantically rich fused outputs [8]. This semantic awareness sets LLM-based approaches apart from traditional fusion methods. Studies in multi-focus image fusion that explores combination of deep networks with edge-preserving techniques [9], and residual feature learning models have improved focus discrimination in self-supervised settings [10]. The use of visual salience priors has also contributed to enhanced focus-aware fusion [11]. In the medical domain, hybrid CNN-based methods incorporating non-subsampled contourlet transforms have improved structural fidelity [12], while dual-domain cross-attention mechanisms combined with 3D manifold fitting have further advanced fusion quality in multi-modal medical imaging [13]. Collectively, these works underscore the growing role of LLMs in elevating the semantic and contextual understanding in image fusion tasks.

This section presents a literature survey on various LLM-driven recommendation models, highlighting their methodologies, evaluation metrics, datasets, and key shortcomings. The study covers models ranging from transformer-based approaches such as BERT4Rec [14] and Transformer4Rec [15] to meta-learning and multimodal fusion techniques, as shown in Table 1. By analyzing a wide range of models—including GPT-Rec [16], CORE [17], SASRec [18], LLM4Rec [19], Transformer4Rec [15], GraphLLM-Rec [20], XRecLLM [21], T5-Rec [22], HybridLLM-Rec [23], RecGNN-LLM [20], DeepLLMRec [24], ZeroShotRec [25], MetaRec-LLM [25], PromptRec [26], and FairLLM-Rec [27]—this study identifies emerging trends, challenges, and potential research directions for advancing recommendation systems using large language models (LLMs) and deep learning architectures.

Wu et al. [28] surveys Large Language Model (LLM)-based recommendation systems, categorizing them into Discriminative and Generative LLMs for Recommendation (DLLM4Rec and GLLM4Rec). This paper analyzes existing systems, identifies key challenges, and provides valuable insights to advance the field.

Table 1
Literature survey on recommendation systems using large language models.

Model	Methodology	Evaluation metrics	Datasets	Shortcomings
BERT4Rec	Uses bidirectional transformers for sequential recommendations	NDCG@10, Recall@10, MRR	MovieLens, Amazon	Struggles with cold-start users.
GPT-Rec	Adapts GPT for personalized item recommendations	Precision@10, Recall@20, F1-score	Netflix, Goodreads	High computational cost.
CORE	Contrastive learning for recommendation embeddings	NDCG@50, HR@10	Yelp, Epinions	Requires extensive fine-tuning.
STAR	Training-free approach using LLM-generated embeddings	MRR, Recall@10	MovieLens, Book-Crossing	Struggles with long-tail items.
SASRec	Self-attention-based sequential recommender	HR@20, MAP	LastFM, YouTube	Does not capture item novelty well.
LLM4Rec	Uses pre-trained LLMs for user-item interactions	RMSE, MAE, Recall@10	Amazon Electronics, IMDB	Poor scalability for large datasets.
Transformer4Rec	Leverages transformers for next-item prediction	NDCG@50, Precision@10	Spotify, Netflix	Struggles with sparse datasets.
T5-Rec	Adapts T5 for sequential recommendation tasks	Hit Rate@10, MAP	MovieLens, Yelp	Needs large-scale pretraining.
GraphLLM-Rec	Integrates knowledge graphs with LLMs for recommendations	BLEU, NDCG	Goodreads, Amazon Music	High memory consumption.
HybridLLM-Rec	Combines collaborative filtering with LLM-generated embeddings	Accuracy, Recall@10, MRR	Amazon Beauty, Spotify	Limited interpretability.
XRecLLM	Cross-modal recommendation using multimodal LLM embeddings	Recall@20, MRR, HR@10	Netflix, YouTube, LastFM	Struggles with real-time inference.
RecGNN-LLM	Graph neural network combined with LLM for recommendations	NDCG@20, MAP	Yelp, Amazon Movies	Requires high computational power.
DeepLLMRec	Deep-learning-based hybrid recommendation using LLMs	Precision@20, F1-score, Recall@50	Amazon Music, Goodreads	Struggles with real-time user preferences.
ZeroShotRec	Zero-shot learning for cold-start recommendation via LLM	Recall@10, NDCG@5, BLEU	LastFM, IMDb, Spotify	Struggles with niche recommendations.
MetaRec-LLM	Meta-learning framework for adaptive recommendations	HR@10, NDCG@20, Precision@10	Yelp, Amazon, Goodreads	Requires high training time.
PromptRec	Prompt-based recommendation using LLM zero-shot learning	HR@10, MAP, BLEU	Amazon Music, Goodreads	Sensitive to prompt design.
FairLLM-Rec	Bias-aware LLM-driven recommendation model	Recall@20, NDCG@50, Precision@10	Netflix, Yelp, Amazon Electronics	Struggles with fairness across user groups.

The paper [29] introduces a novel personalized recommendation system using LLMs, achieving significant performance improvements: 8.6% in NDCG@10 and 10.5% in MRR over state-of-the-art baselines. The system integrates LLM-based semantic understanding with user preference modeling, effectively recommending long-tail items and diverse genres.

LLMRS [30], a zero-shot recommender system utilizing LLMs to encode user reviews, outperforms a ranking-based baseline on Amazon product review data by generating more reliable, user-tailored recommendations. Additionally, a Llama-2 [31] LLM-based product recommendation system generates personalized user embeddings and outperforms traditional methods (collaborative/content-based filtering) in click-through and purchase rates on a real-world e-commerce dataset. However, the limitations of traditional systems (e.g., cold start problem and capturing complex preferences) are not explicitly addressed as limitations of this system.

Xu et al. [32] explores the use of LLMs in recommendation systems via prompt engineering. This paper proposes a framework, analyzes the impact of various LLM and prompt engineering factors on performance using two datasets, and highlights promising future research directions. Key research gaps include the effects of LLM architecture, parameter scale, context length, and different prompt components on recommendation quality.

The study [33] proposes a Llama-2-based product recommendation system, addressing the limitations of traditional methods like collaborative and content-based filtering. The research gap identified is the under-exploration of LLMs for personalized product recommendations, particularly in generating user embeddings to improve recommendation accuracy.

OpenP5 [34], an open-source platform for developing and evaluating LLM-based recommender systems, is introduced. A limitation highlighted is the nascent nature of the field and the current lack of similar open-source R & D platforms. Current LLMs underperform in recommendation tasks due to insufficient use of collaborative information. LAMAR [35], a framework combining LLMs with traditional models to leverage both collaborative and semantic data, improves recommendation accuracy.

Current LLM-based recommendation systems are slow and rely on pre-trained knowledge. CherryRec [36], a new framework, addresses

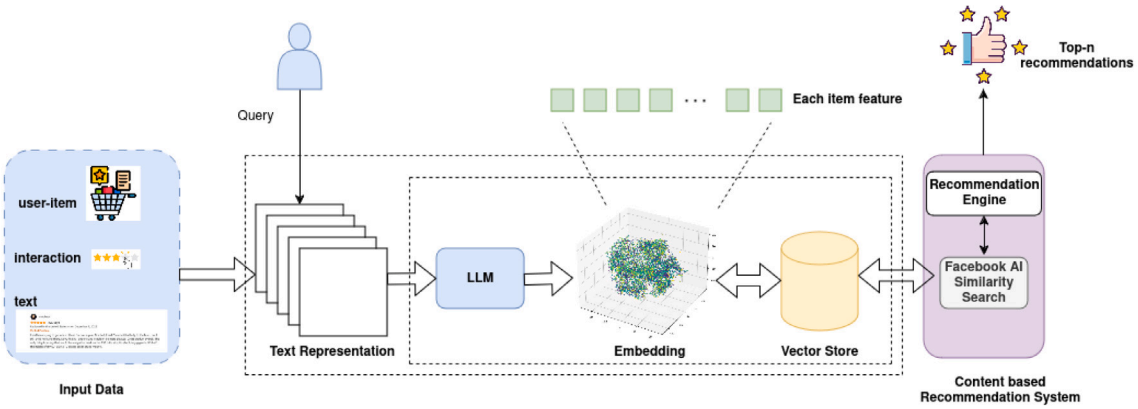

Fig. 1. Proposed recommendation pipeline.

this issue by using a three-stage process (rapid selection, LLM evaluation, and scoring) to improve both the speed and accuracy of news recommendations.

Research gaps exist in applying Federated Learning to LLM-based recommendation systems, particularly in client performance imbalance and high client-side computational/storage demands. The PPLR framework [37] addresses these challenges through dynamic parameter aggregation and selective model offloading to improve performance and privacy.

The study [24] identifies research gaps in traditional collaborative filtering's handling of cold start and data sparsity problems. It proposes and validates a hybrid recommendation system combining collaborative filtering with LLMs to enhance accuracy and diversity, addressing these limitations.

LLM-based recommendation systems using Direct Preference Optimization (DPO) [38] suffer from a bias towards popular items. The paper introduces SPRec, a self-play framework that mitigates this bias by using model predictions as negative samples during training, improving recommendation accuracy and fairness without extra data.

Most research focuses on enhancing recommendation systems with textual data, while the paper [39] presents LLM4IDRec, which augments ID-based recommendation data (lacking textual data), improving performance without modifying existing recommendation models. The key research gap addressed is the under-exploration of LLMs for ID-only recommendation data. While much of the research focuses on improving recommendation systems, there is limited investigation into improving embedding generation using LLMs.

3. Problem definition

Given a dataset:

$$\mathcal{D} = \{(p_1, t_1), (p_2, t_2), \ldots, (p_n, t_n)\},$$

where p_i represents the ith item and t_i denotes its corresponding textual representation – comprising item metadata, descriptions, and user-generated reviews – the objective is to learn a mapping function:

$$f : t \to \mathbb{R}^d$$

that projects each item's textual representation t_i into a d-dimensional semantic space. The embedding function f is optimized to ensure that semantically similar products are positioned closer in this space, while dissimilar products are placed farther apart.

The generated embeddings are indexed within a searchable structure $\mathcal{I}$, enabling efficient similarity-based retrieval. Given a query embedding $e_q = f(t_q)$, the system retrieves the top-k most relevant products:

$$\{p_{r_1}, p_{r_2}, \ldots, p_{r_k}\}$$

such that the similarity measure $s(e_q, e_r)$ is maximized, where $e_r = f(t_r)$ represents the embedding of a retrieved product.

This approach ensures that the retrieval mechanism effectively identifies and ranks items based on their semantic relevance, enhancing the quality and accuracy of recommendations.

4. Methodology

4.1. Model overview

The proposed recommendation pipeline of the model is depicted in Fig. 1. The proposed system begins by preprocessing the dataset to generate a textual representation for each item. This representation consolidates essential item details into a single text-based format. These representations serve as input for generating high-dimensional embeddings using Large Language Models such as LLaMA2 [40], Mistral [41]and Phi-3-mini [42].

The embeddings capture nuanced semantic relationships among the products, enabling the system to form a robust foundation for similarity-based recommendations. To enable scalable recommendations, the embeddings are indexed using FAISS (Facebook AI Similarity Search), an efficient similarity search library optimized for large-scale datasets. The FAISS index is constructed with an L2 distance metric, ensuring fast and accurate retrieval of top-k similar products for any given query. By integrating this index with the embedding model, the system can dynamically generate recommendations for new or unseen items, effectively addressing the cold-start problem.

4.2. Text representation creation

Each item in the dataset is extracted and converted to its text representation. The text representation of each item is a structured string concatenating the details in the dataset corresponding to each row. This representation is essential for providing meaningful inputs to the embeddings. This will help to ensure the capture of semantic details relevant to the recommendations. The metadata for each row in the dataset is aggregated into a textual format. Mathematically, this can be represented as:

$$\mathcal{T}_i = f(M_i) = \text{Concatenate}(M_i^1, M_i^2, \ldots, M_i^n),$$

where M_{ij} represents the jth metadata field of the ith row.

4.3. Embedding model selection and usage

In this study, we selected LLaMA2 [40], Mistral [41]and Phi-3 mini [42] for embedding generation, due to their strong balance between the quality of semantic representation and computational efficiency. Unlike heavier models such as BERT [43] or OpenAI's larger

Table 2
Description of variables and their corresponding meanings.

Variable	Description
S_i^j	jth metadata field of the ith item.
T_i	Text representation of the ith item.
E_i	High-dimensional embedding vector of the ith item generated using a pre-trained LLM.
X	Embedding matrix representing all item embeddings.
N	Total number of items in the dataset.
R	Overall relevance score of an item.
S_r	Randomly selected item for comparison.
E_r	High-dimensional embedding vector of the randomly selected item.
R_g	Genre-based relevance score.
R_{d_e}	Description-based relevance score.
d	Dimensionality of the embedding space.

GPT-based models [44], these lightweight models offer faster inference speeds and lower resource consumption, making them ideal for scalable recommendation systems where latency is a critical factor.

Additionally, these models are capable of capturing fine-grained semantic relationships, even without extensive domain-specific fine-tuning. This feature aligns with the system's goal of generalization across diverse recommendation domains. We used the pre-trained configurations of these models as it-is, without additional fine-tuning for the embedding generation. This approach was adopted to assess the inherent generalization capability and to maintain reproducibility of our experimental results.

4.4. Embedding generation

Each text representation T_i is encoded into a high-dimensional embedding vector E_i using a pre-trained large language model (LLM). Models such as LLaMa2, Mistral, and Phi-3 mini are utilized in the proposed work. The LLM projects T_i into a semantic space as follows:

$$E_i = \text{Model}(T_i)$$

- Model represents the specific LLM employed (e.g., LLaMa2, Mistral, or Phi-3 mini).
- $E_i \in \mathbb{R}^d$ is the embedding vector with dimension d.

These embeddings are stored in a NumPy array for computational efficiency.

4.5. Modeling paradigm

Given a dataset of items $T = \{T_1, T_2, \ldots, T_N\}$, where each T_i represents a textual description, embeddings are generated using a local embedding API. The process is outlined as follows:

4.5.1. Prompt to embedding API

Each item T_i is sent as a prompt to a local embedding API of LLM (e.g., LLaMA2), which returns a corresponding embedding $\mathbf{E}_i \in \mathbb{R}^d$, where d is the dimensionality of the embedding space. The embedding retrieval process is formalized as:

$$E_i = \text{EmbeddingAPI}(T_i),$$

where EmbeddingAPI represents the function that maps the textual input to a dense numerical representation.

For implementation, this is achieved using the following HTTP request:

```
res = requests.post('http://localhost:11434/api/embeddings',
json={'model': 'llama2', 'prompt': representation})
embedding = res.json()['embedding']
```

4.5.2. Embeddings assembly

The embeddings $E_1, E_2, \ldots, E_N$ are collected to form a matrix $E \in \mathbb{R}^{N \times d}$, where:

$$E = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}.$$

In this matrix:

- N: Total number of items in the dataset.
- d: Dimensionality of the embedding space.

4.5.3. Dataset representation

The matrix E provides a structured numerical representation of the entire dataset, suitable for machine learning tasks such as similarity search or recommendation systems.

4.6. Facebook AI similarity search (FAISS) based index construction

To facilitate efficient similarity search, FAISS is employed to index the semantic embeddings. FAISS leverages the k-nearest neighbors (k-NN) algorithm, which allows for fast and scalable retrieval of similar items. The semantic embeddings, denoted by X, are added to the FAISS index using the operation:

$$\text{Index.add}(X),$$

where $X \in \mathbb{R}^{N \times d}$ is the matrix of embeddings, and N represents the number of items, with d being the dimensionality of the embeddings.

4.6.1. Similarity metric

The similarity metric utilized for the search is the squared Euclidean distance, defined as:

$$d^2(x, y) = \sum_{i=1}^{d} (x_i - y_i)^2,$$

where $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$ are vectors in $\mathbb{R}^d$, and d denotes the dimensionality of the vectors.

4.6.2. Querying the FAISS index

For a given query embedding E_q, the top-k most similar items are retrieved by searching the FAISS index:

$$D, I = \text{Index.search}(E_q, k), \tag{1}$$

where:

- $D \in \mathbb{R}^k$ contains the distances to the k-nearest neighbors.
- $I \in \mathbb{Z}^k$ contains the indices of the k-nearest neighbors.

4.7. Recommendation workflow

To generate recommendations, a randomly selected item S_r is used as a query, and its corresponding embedding E_r is employed to find the most similar embeddings. Using FAISS, the top-k-nearest neighbors are identified:

$$D, I = \text{FAISS.search}(E_r, k), \tag{2}$$

where:

- D: The distances to the k-nearest neighbors.
- I: The indices of the k-nearest neighbors within the dataset.

4.7.1. Relevance definition

In the context of a recommendation system, relevance is defined as the measure of how well a recommended item aligns with the importance or similarity criteria based on a specific query item. In this work, two types of relevance are considered: genre-based relevance and description-based relevance. These definitions are tailored to accommodate datasets from diverse domains.

- Genre-based Relevance (R_g): If the dataset includes a genre attribute, two items are considered genre-relevant if they belong to the same genre or category. This relevance is defined as:

$$R_{\text{g}} : \text{Relevance}_{\text{genre}}(S_{\text{i}}, S_{\text{r}}) = \begin{cases} 1 & \text{if Genre}(S_{\text{i}}) = \text{Genre}(S_{\text{r}}), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

  This binary definition simplifies relevance computation for structured datasets; however, it is acknowledged that in real-world scenarios, genre similarity may exist on a continuum (e.g., "Action" and "Adventure" genres are more related than "Action" and "Romance"). In this work, for simplicity and consistency in evaluation across different domains, the binary approach to genre-relevance definition is adopted, especially, where detailed genre similarity data is not available. Future research could extend this framework by using more sophisticated approaches such as hierarchical taxonomies, knowledge graphs, which could capture such nuanced relationships.
- Description-based Relevance (R_{de}): Items are considered description-relevant if their embeddings reflect semantic similarity in textual descriptions. This similarity is quantified using cosine similarity between the embeddings $\mathbf{E}_i$ and $\mathbf{E}_r$ as follows:

$$R_{\text{de}} = \text{Similarity}(S_{\text{i}}, S_{\text{r}}) = \frac{\mathbf{E}_{\text{i}} \cdot \mathbf{E}_{\text{r}}}{\|\mathbf{E}_{\text{i}}\| \cdot \|\mathbf{E}_{\text{r}}\|}, \tag{4}$$

  where $\mathbf{E}_i$ and $\mathbf{E}_r$ are the embeddings of the items S_i and S_r, respectively.
  The top-k items with the highest similarity scores are considered description-relevant.

4.7.2. Overall relevance

The overall relevance (R) is computed as the union of genre-relevant and description-relevant items. This combined set of relevant items is then used as the ground truth for evaluating recall at various thresholds:

$$R = R_{\text{g}} \cup R_{\text{de}}. \tag{5}$$

4.8. Recommendation generation using FAISS

To generate recommendations using the FAISS index, the query embedding is searched against the index with the following command:

$$D, I = \text{index\_amazon.search}(\textbf{embedding}, \text{top\_n}), \tag{6}$$

where D represents the distances, and I contains the indices of the top-n nearest neighbors.

Subsequently, the recommended product IDs are extracted using:

$$\text{recommended\_ids} = [\text{df.iloc}[i][\text{'product\_id'}] \text{ for } i \text{ in } I.\text{flatten}()]. \tag{7}$$

The description of the variables used and its interpretations are depicted in Table 2.

5. Experiments

In this study, a series of experiments were conducted to evaluate the performance of the proposed recommendation models across multiple datasets, spanning diverse domains such as entertainment, e-commerce, music, and literature. The following research questions guide the evaluation and analysis of the experimental results.

RQ1: How does the performance of the proposed models compare with baseline models?

RQ2: What is the impact of the embedding generation method on the performance of the proposed models in enhancing recommendation systems?

RQ3: How do the proposed models perform in terms of recall at different cut-off points for datasets with varying levels of sparsity and data distribution?

5.1. Experimental setup

To evaluate the proposed system, the dataset D comprising textual descriptions of items are preprocessed to ensure consistency and remove noise. Each item is represented textually by aggregating relevant attributes, such as title, description, and category. These representations are fed into the pre-trained embedding model M, accessed via API, to generate semantic embeddings of dimensionality d. The embeddings are stored in a matrix $X \in \mathbb{R}^{|D| \times d}$, where each row corresponds to a product in D.

The matrix X is indexed using FAISS [45], a high-performance similarity search library, employing the `IndexFlatL2` index type, which provides efficient nearest-neighbor retrieval. To ensure reproducibility, the embedding matrix X and the FAISS index I are saved for subsequent experiments.

For evaluation, a subset of items from D is randomly selected as query products p_{query}. Their embeddings e_{query} are generated using M and queried against the FAISS index I to retrieve the top-N most similar products. The recommendation quality is assessed using metrics such as precision@N, recall@N, and Normalized Discounted Cumulative Gain (NDCG).

The experimental setup includes comparisons between different embedding models (e.g., transformer-based or traditional word embeddings) and FAISS indexing methods, allowing insights into their respective contributions to system performance. Hyperparameters such as embedding dimensionality, FAISS configuration, and N are tuned via grid search to achieve optimal results.

To ensure a fair comparison and to analyze the impact of different large language models (LLMs) on the recommendation task, we consider three model variants that are similar in scale (with the exception of Phi-3-mini, which is lighter). The details of these models are summarized in Table 3.

5.1.1. Model variants

The three proposed model variants differ only in the choice of embeddings used, to generate semantic representations, while the proposed architecture and the methodology of the models remain the same

- **Proposed Model A:** This model uses embeddings from the LLaMA2-7B model, which provides rich and high-capacity semantic representations.

Table 3
Comparison of model scales.

Model	Parameters	Approximate FLOPs	Comments
LLaMA2-7B	7B	$\sim 1.4 \times 10^{12}$	Standard model, open weights
Mistral-7B	7B	$\sim 1.2 \times 10^{12}$	Dense/sparse mixture design
Phi-3-mini	3.8B	$\sim 0.7 \times 10^{12}$	Lightweight, optimized for efficiency

Table 4
Summary of datasets, their categories, and key statistics.

Dataset	Category	Statistics
Amazon Beauty	E-commerce	Users: 10M, Products: 50K, Reviews: 200K
Yelp	Local Business	Users: 2M, Businesses: 50K, Reviews: 5M
LastFM	Music Streaming	Users: 3M, Tracks: 2M, Interactions: 100M
MovieLens	Entertainment	Users: 1M, Movies: 10K, Ratings: 10M

- **Proposed Model B:** This model uses embeddings from the Mistral-7B model, which balances efficiency and semantic richness through architectural optimizations.
- **Proposed Model C:** This model uses embeddings from the Phi-3-mini model, a lightweight and efficient model suitable for resource-constrained scenarios.

5.1.2. Datasets

For the experimental evaluation, several diverse datasets were utilized to validate the effectiveness of the proposed recommendation system. The dataset categories summarized in Table 4 highlight the diverse domains considered in our study, ranging from entertainment and media to e-commerce, local business reviews, and music.

For clarity, the datasets are categorized as follows:

- **Structured Data:** The datasets: MovieLens [46], LastFM [47] primarily consist of well-organized user-item interaction matrices with explicit ratings.
- **Semi-Structured Data:** The datasets: Amazon Beauty [48], Yelp [49] contain both structured numerical ratings and unstructured textual information such as product descriptions or user reviews.

This categorization provides information on the nature of the datasets and helps in analyzing their impact on the performance of the proposed recommendation models. These datasets collectively ensure a robust evaluation across diverse domains, showcasing the adaptability of the semantic embedding approach. By employing multiple datasets, the system's performance is thoroughly validated, demonstrating its ability to cater to varied user preferences and application contexts.

5.1.3. Evaluation metric

To evaluate the recommendation system, several widely used metrics were employed to quantify the quality of the recommendations. The first metric, Precision@N, measures the proportion of recommended items in the top-N that are relevant to the query product. It is defined as:

$$\text{Precision@}N = \frac{|\text{Relevant Items} \cap \text{Recommended Items}|}{N} \tag{8}$$

This metric focuses on the accuracy of the recommendations, providing insights into the system's ability to return relevant results.

The second metric, Recall@N, evaluates the proportion of relevant items successfully retrieved in the top-N recommendations. It is calculated as:

$$\text{Recall@}N = \frac{|\text{Relevant Items} \cap \text{Recommended Items}|}{|\text{Relevant Items}|} \tag{9}$$

Recall@N highlights the system's capacity to identify all relevant products, emphasizing comprehensiveness over precision.

The third metric, F_1-Score@N, is the harmonic mean of Precision@N and Recall@N, balancing their contributions. It is computed as:

$$F_1\text{-Score@}N = 2 \cdot \frac{\text{Precision@}N \cdot \text{Recall@}N}{\text{Precision@}N + \text{Recall@}N} \tag{10}$$

The F_1-Score provides a single measure to assess the trade-off between precision and recall.

Finally, the Normalized Discounted Cumulative Gain (NDCG@N) evaluates the ranking quality of the recommendations. It is defined as:

$$\text{NDCG@}N = \frac{1}{Z} \sum_{i=1}^{N} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{11}$$

where rel_i is the relevance score of the item at rank i, and Z is a normalization factor to ensure that the maximum possible NDCG@N is 1. NDCG@N accounts for both the relevance and order of recommendations, making it a comprehensive metric for evaluating ranked lists.

These metrics collectively provide a holistic evaluation of the recommendation system, capturing different aspects such as precision, recall, ranking quality, and overall performance.

Additionally, to evaluate the real-world performance of the recommendation system, several widely used business-relevant evaluation measures are also employed. The first metric, Coverage measures the proportion of items from the catalog that the recommendation system is capable of recommending. It is defined as:

$$\text{Coverage} = \frac{\text{Number of unique items recommended}}{\text{Total number of items in the dataset}} \times 100 \tag{12}$$

The second metric, Diversity quantifies how different the recommended items are from each other. It is computed using the equation below:

$$\text{Diversity} = \frac{1}{\text{N(N}-1)} \sum_{i \neq j} (1 - \text{Similarity}(i, j)) \tag{13}$$

The third metric, Novelty measures how unexpected or new the recommended items are to the user. Novelty can be computed by evaluating the popularity of the recommended items, with lower popularity indicating higher novelty.

$$\text{Novelty} = \frac{1}{\text{N}} \sum_{i=1}^{\text{N}} \text{Popularity}(i) \tag{14}$$

Finally, the Serendipity evaluates how surprisingly relevant the recommended items are. It can be computed as:

$$\text{Serendipity} = \frac{\text{Relevance} \times (1 - \text{Popularity})}{\text{Total Recommendations}} \tag{15}$$

These business-relevant metrics provide additional insights into the system's ability to offer variety, expose users to new content, and ensure engaging recommendations.

5.1.4. Baselines

To evaluate the performance of the proposed LLM based recommendation model, it is compared first with straightforward fundamental baseline models, which include the following:

- BaseMF [50]: Base Matrix Factorization (BaseMF) is a foundational collaborative filtering approach that relies on decomposing the user-item interaction matrix into latent factors. These factors capture the underlying preferences of users and characteristics of items, enabling predictions for unobserved interactions.

BaseMF's simplicity and effectiveness make it a common baseline for recommendation systems.

- NCF [51]: Neural Collaborative Filtering (NCF) extends traditional matrix factorization methods by incorporating neural networks to model user-item interactions. By learning non-linear and complex relationships, NCF enhances predictive performance. It uses multi-layer perceptrons (MLPs) to learn a latent feature space, enabling flexible and expressive modeling.
- AutoRec [52]: AutoRec is an autoencoder-based collaborative filtering model designed for recommendations. It treats the user-item interaction matrix as input to an autoencoder, where the encoder maps input to a latent space, and the decoder reconstructs missing interactions. AutoRec can be user-based or item-based and excels at capturing low-dimensional representations for recommendation tasks.
- GCMC [53]: GCMC integrates graph convolutional networks (GCNs) into recommendation systems to model user-item interactions as a bipartite graph. By propagating and aggregating information across the graph, GCMC captures structural relationships and high-order dependencies, improving performance in sparse datasets.
- PinSage [54]: PinSage is a graph-based recommendation model that combines graph neural networks (GNNs) with efficient sampling techniques for large-scale recommender systems. It processes a graph of user-item interactions, learning node embeddings that capture contextual and structural information for personalized recommendations.
- NGCF [55]: NGCF is a GNN-based model that explicitly incorporates higher-order connectivity into the recommendation process. By propagating and updating user and item embeddings through graph convolutions, NGCF models the collaborative signal in a multi-hop neighborhood, enhancing recommendation quality.
- STGCN [56]: STGCN extends graph-based recommendation models by incorporating temporal dynamics into the learning process. It captures both graph structural information and temporal patterns in user-item interactions, making it suitable for time-sensitive recommendations.
- LightGCN [57]: LightGCN simplifies traditional GCN-based recommendation models by removing unnecessary components like feature transformations and nonlinear activation functions. This lightweight design focuses on propagating and aggregating embeddings, providing a more efficient yet effective approach for collaborative filtering tasks.

To ensure a fair comparison, we also evaluate our approach against several well-established sequential recommendation models that rely solely on user interaction data also. The baseline models for sequential recommendation considered for comparison are as follows:

- Caser [58]: Treats sequential recommendation as a Markov Chain problem and applies Convolutional Neural Networks (CNNs) to capture user behavior patterns.
- HGN [59]: Employs hierarchical gating networks to model user preferences by integrating both long-term and short-term behaviors.
- GRU4Rec [60]: Utilizes Gated Recurrent Units (GRU) ? to encode user interaction sequences effectively.
- BERT4Rec [14]: Adopts a BERT-style masked language modeling approach to learn bidirectional representations for sequential recommendations.
- FDSA [61]: Leverages self-attention mechanisms to model feature sequences for improved recommendation performance.
- SASRec [18]: Implements a self-attention framework to capture dependencies within sequential user interactions.

This selection of baselines ensures a comprehensive comparison by covering various modeling approaches, including CNNs, recurrent networks, self-attention mechanisms, and transformer-based architectures.

Additionally, to ensure a fair comparison, we compare our model against the following recent embedding models.

- GTE-small [62]: GTE-small is a general purpose text embedding model, which uses a multi-stage approach to capture semantic representations, while maintaining computational efficiency.
- E5-small [63]: E5-small is a weakly supervised contrastive embedding model designed to enhance retrieval and classification tasks, utilizing a more compact architecture.
- BGE-small [64]: BGE-small utilizes a verification-based approach for embedding generation offering high quality representations.

5.1.5. Parameter configuration

The parameter configuration in this recommendation system is carefully designed to evaluate the performance of embeddings generated by three distinct large language models (LLMs): LLaMA 2, Mistral, and Phi-3 mini. Each of these models produces high-dimensional embeddings, capturing nuanced semantic and contextual information about the items in the dataset. The embedding dimension (`dim`) is set to 4096 for LLaMA 2 and Mistral, and 3072 for Phi-3 mini, ensuring that the differences in model configurations are taken into account while maintaining comparability of results across the models.

For retrieval, a `FAISS IndexFlatL2` structure is employed. This index type helps to perform exhaustive search using L2 (Euclidean) distance, ensuring that the retrieval quality reflects the closest match based on the embedding space, without any approximation errors. Since `IndexFlatL2` is a brute-force index, it does not require hyperparameter settings such as clustering parameters (`nlist`, `nprobe`) or graph connectivity parameters (`M`, `efSearch`). But inorder to decide on the index type, a fair comparison is made between the FAISS index type L2 and Inner Product (IP), during performance analysis depicted in Table 10. Based on the investigation, most suitable index type `FAISS IndexFlatL2` is chosen for the proposed work. This choice prioritizes maximum recall and experimental reproducibility.

In relevance simulation, the system considers two aspects to define the relevance set: genre alignment and description similarity. Genre alignment filters items that share the same genre as the randomly selected item, while description similarity is calculated based on cosine similarity between item embeddings. Although FAISS uses L2 distance for retrieval, cosine similarity is separately computed during evaluation to better assess semantic proximity between items.

The parameter `top_n`, representing the number of top recommendations retrieved during the search, is varied across 10, 20, 40, 60, and 80. This variability allows a comprehensive evaluation of recall at different levels, enabling a better understanding of how effectively the models retrieve relevant recommendations. In relevance simulation, the system considers both genre alignment and description similarity to define the relevance set. Genre alignment filters items that share the same genre as the randomly selected item, while description similarity is determined by calculating the cosine similarity of embeddings generated by the respective LLMs. By leveraging embeddings from LLaMA2, Mistral, and Phi-3 mini, the system evaluates the impact of different language models on recommendation quality. Recall metrics are computed for each `top_n` value, offering a detailed comparison of the three LLMs' effectiveness in retrieving semantically relevant and genre-consistent recommendations. This multi-model approach allows for a deeper exploration of embedding quality and its influence on recommendation system performance.

5.2. Performance analysis on recommendation system

The performance of the proposed model is investigated from different perspective to answer the research questions: RQ1, RQ2 and

RQ3. Initially, the text representations are visualized to understand the qualitative perspective on the distribution of key terms. Then the performance of the model is evaluated statistically and theoretically, to analyze the ranking quality of the recommendation system.

5.2.1. Visualizing text representations with word clouds

To gain deeper insights into the textual representations used for generating semantic embeddings, word cloud visualizations were employed across different datasets. The most frequent terms using word clouds are visualized as shown in Fig. 2 presents word clouds for three distinct datasets: the MovieLens (ML-1M) dataset, the LastFM dataset, and the Yelp dataset. The first image (left) shows the word cloud for movie-text representations in the MovieLens dataset, highlighting the prominent terms and keywords related to the movies. The second image (right) represents the track-text features from the LastFM dataset, showcasing the most common words associated with music tracks and their metadata. Finally, the third image (center) visualizes the review-text representations from the Yelp dataset, emphasizing the frequent terms found in user reviews. These visualizations provide a qualitative perspective on the distribution of key terms and their relative importance in the textual descriptions, contributing to the interpretation of the experimental results.

The relevance of these word clouds to the research questions is inferred as follows:

- **Addressing RQ1:** Word clouds help illustrate the diversity and richness of textual content within each dataset. A higher density of meaningful terms suggests that the dataset provides more informative textual features, which may contribute to the superior performance of the proposed model over traditional baselines. Conversely, sparsely informative text may explain performance limitations in certain datasets.
- **Addressing RQ2:** Since the proposed model derives semantic embeddings from textual descriptions, the word clouds offer a qualitative analysis of the input text structure. The presence of semantically rich and diverse vocabulary enhances the quality of the embeddings, leading to improved recommendation accuracy. In contrast, datasets with repetitive or limited vocabulary may result in weaker embeddings, affecting the overall performance.
- **Addressing RQ3:** By visualizing the word distributions, it is possible to infer dataset sparsity and the informativeness of textual descriptions. Datasets with dense, diverse textual content are likely to yield more discriminative embeddings, positively impacting recall scores at lower cut-offs. Conversely, datasets with sparse or highly redundant textual representations may exhibit lower performance due to weaker semantic differentiation.

Thus, word cloud visualizations serve as a valuable interpretability tool, offering a qualitative understanding of dataset characteristics and their implications on embedding quality and recommendation effectiveness. This analysis reinforces the significance of textual feature richness in shaping the performance of embedding-driven recommendation models.

5.2.2. Statistical evaluation of model performance

The performance of the proposed models is assessed by comparing them against both traditional recommendation models and sequential recommendation baselines and embedding-based models. Three variants of the proposed model, namely Proposed Models A, B, and C, each differing based on the choice of embeddings used for enhancing recommendation tasks are developed. Experiments are conducted across different dataset categories to evaluate the effectiveness of these variants.

Fig. 2. Word cloud for different datasets.

- **Performance of Proposed Models with Straightforward Recommendation Baselines**
  The comparative performance analysis of the proposed models across dataset categories is presented in Table 5. The results indicate a substantial improvement in recommendation accuracy achieved by the proposed methods. Across all evaluation metrics (Recall@20, NDCG@20, Recall@40, and NDCG@40), the proposed models (A, B, and C) consistently surpass the baseline models, demonstrating their ability to capture richer semantic relationships.
  Notably, the improvement is more pronounced in the semi-structured Yelp dataset, where conventional collaborative filtering and graph-based methods struggle due to the inherent sparsity and noise in user-item interactions. In contrast, the ML-IM dataset, being more structured, allows baseline models to perform reasonably well; however, the proposed models still exhibit superior performance, showcasing their robustness across varying data structures. The consistent boost in NDCG scores further suggests that the proposed models are more effective in ranking relevant items, making them highly suitable for real-world recommendation systems dealing with both structured and semi-structured data.

  - **Addressing RQ1:** The evaluation results demonstrate that the proposed models (A, B, and C) consistently outperform the baseline models across all evaluation metrics. This superior performance highlights the ability of the proposed models to capture richer semantic relationships in recommendation tasks, thereby overcoming the limitations of traditional and sequential recommendation models.
  - **Addressing RQ2:** The observed variation in performance across the three proposed model variants underscores the critical role of embedding generation methods in recommendation effectiveness. The experimental results suggest that the choice of embedding strategy directly influences the model's capability to retrieve and rank relevant items. This finding highlights the necessity of carefully selecting embedding techniques to optimize recommendation quality.
  - **Addressing RQ3:** The comparative analysis of structured (ML-IM) and semi-structured (Yelp) datasets reveals that the proposed models exhibit substantial improvements across both data structures. While the ML-IM dataset enables relatively strong baseline model performance, the proposed models still achieve superior results, demonstrating their adaptability. The performance gains are more pronounced in the semi-structured Yelp dataset, where the proposed models effectively mitigate the impact of sparsity and noise—challenges typically faced by conventional collaborative filtering and graph-based approaches. Furthermore,

Table 5
Performance comparison of recommendation models on Yelp and ML-1M datasets. The best-performing values for each metric are highlighted in **bold**.

Dataset	Metric	Baseline models								Proposed models		
		BiasMF	NCF	AutoR	GCMC	PinSage	NGCF	STGCN	LightGCN	A	B	C
Yelp	Recall@20	0.0190	0.0252	0.0259	0.0266	0.0345	0.0294	0.0309	0.0482	**0.5000**	**0.5123**	**0.3300**
	NDCG@20	0.0161	0.0202	0.0210	0.0251	0.0288	0.0243	0.0262	0.0409	**0.6131**	**0.6131**	**0.4693**
	Recall@40	0.0371	0.0487	0.0504	0.0585	0.0599	0.0522	0.0504	0.0803	**0.5200**	**0.7613**	**0.4900**
	NDCG@40	0.0227	0.0289	0.0301	0.0373	0.0385	0.0330	0.0332	0.0527	**0.6131**	**0.6131**	**0.5360**
ML-IM	Recall@20	0.0196	0.0110	0.0239	0.0301	0.0576	0.0552	0.0369	0.0985	**0.5000**	**0.5000**	**0.5000**
	NDCG@20	0.0105	0.0153	0.0198	0.0224	0.0331	0.0302	0.0250	0.0551	**0.6131**	**0.6131**	**0.4693**
	Recall@40	0.0375	0.0243	0.0485	0.0526	0.0853	0.0815	0.0627	0.1250	**0.5200**	**0.7613**	**0.4900**
	NDCG@40	0.0256	0.0298	0.0321	0.0389	0.0452	0.0417	0.0363	0.0665	**0.6131**	**0.6131**	**0.5360**

Table 6
Performance comparison with baselines for sequential recommendation.

Methods	ML-1M		Amazon		LastFM	
	HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
Caser	0.0912	0.0565	0.0205	0.0131	0.0303	0.0178
HGN	0.1430	0.0874	0.0325	0.0206	0.0321	0.0175
GRU4Rec	0.0806	0.0475	0.0164	0.0099	0.0275	0.0158
BERT4Rec	0.1308	0.0804	0.0203	0.0124	0.0422	0.0269
FDSA	0.1167	0.0762	0.0267	0.0163	0.0303	0.0219
SASRec	0.1078	0.0681	0.0387	0.0249	0.0505	0.0331
Proposed Model A	**0.2012**	**0.1212**	**0.3111**	**0.0922**	0.0604	0.0219
Proposed Model B	0.1152	0.0823	**0.0531**	**0.0823**	**0.1152**	**0.0823**
Proposed Model C	0.1420	0.0687	**0.1420**	**0.0687**	**0.1420**	**0.0687**

the consistent enhancement in NDCG scores suggests that the proposed models rank relevant items more effectively, reinforcing their suitability for diverse recommendation scenarios.

The selection of both structured (ML-IM) and semi-structured (Yelp) datasets ensures a comprehensive evaluation of the proposed models across different data environments. Structured datasets, with their well-defined schemas and explicit user-item interactions, provide a controlled setting to benchmark performance against traditional recommendation techniques. In contrast, semi-structured datasets introduce challenges such as sparsity and noise, requiring more advanced embedding-based approaches to extract meaningful relationships. By evaluating across these two categories, the study demonstrates the adaptability and robustness of the models in handling varying data complexities.

- **Performance of Proposed Models with Sequential Recommendation Baselines**
  Table 6 presents a comparative analysis of various sequential recommendation models across three datasets—ML-1M, Amazon, and LastFM: evaluating their performance based on Hit Rate (HR@5) and Normalized Discounted Cumulative Gain (NDCG@5). The best performing values are highlighted in bold. The results indicate that the proposed models outperform traditional baselines, demonstrating their effectiveness in capturing sequential dependencies and user preferences. Specifically, Proposed Model A achieves the highest performance on the ML-1M and Amazon datasets, suggesting its superior ability to model structured interactions and handle long-tail distributions. Meanwhile, Proposed Model B excels in the LastFM dataset, indicating its robustness in music recommendation tasks where user behavior may exhibit different sequential patterns. The baseline models, such as Caser and GRU4Rec, perform significantly lower, highlighting their limitations in capturing long-term dependencies. Although transformer-based models like BERT4Rec and SASRec show competitive results, they still fall short of the proposed models, suggesting that the novel architectural enhancements in the proposed models contribute to better ranking and recommendation quality. These findings justify the adoption of the proposed approach and suggest potential avenues for further research, such as analyzing dataset-specific performance variations and conducting ablation studies to isolate the most impactful design choices.

  – **Addressing RQ1:** The evaluation results indicate that the proposed models (A, B, and C) outperform the baseline models across most evaluation metrics. In the structured dataset (ML-1M), Proposed Model A achieves the highest HR@5 (0.2012) and NDCG@5 (0.1212), surpassing all baselines. Similarly, in the semi-structured datasets (Amazon and LastFM), the proposed models show substantial improvements, particularly in HR@5 and NDCG@5. The notable performance gain demonstrates the effectiveness of the proposed approach in learning meaningful representations for recommendation tasks.
  – **Addressing RQ2:** The observed variations in performance among Proposed Models A, B, and C highlight the impact of different embedding strategies. Proposed Model A consistently achieves the highest scores in ML-1M and Amazon datasets, suggesting that its embedding generation method better captures user-item interactions. In contrast, Proposed Model B demonstrates superior performance in the LastFM dataset, indicating that different embedding strategies may be more suitable depending on the dataset structure and sparsity. This result underscores the importance of optimizing embedding generation techniques to enhance recommendation quality.
  – **Addressing RQ3:** The results show that the proposed models adapt well to both structured (ML-1M) and semi-structured (Amazon, LastFM) datasets. In ML-1M, where user-item interactions are relatively dense, all proposed models outperform the baselines, demonstrating their effectiveness in structured environments. In the semi-structured Amazon dataset, Proposed Model A exhibits a significant performance improvement, achieving HR@5 of 0.3111 and NDCG@5 of 0.0922, indicating robustness against sparsity and noise. Similarly, in the LastFM dataset, Proposed Model B achieves the highest HR@5 (0.1152) and NDCG@5 (0.0823), suggesting adaptability to different interaction patterns. The results confirm that the proposed

Table 7
Performance comparison of baseline embedding models and the proposed models on the Yelp dataset.

Model	Recall@10	NDCG@10
BGE-small [64]	0.370	0.512
E5-small [63]	0.320	0.490
GTE-small [62]	0.280	0.470
Proposed Model A	**0.400**	**0.590**
Proposed Model B	0.200	0.080
Proposed Model C	0.330	0.469

models effectively address sparsity and distribution challenges, making them suitable for diverse recommendation scenarios.

- **Performance of Proposed Models Compared with Embedding-Based Baselines**
  In this section, the performance of the model are compared with the recent baseline embedding models. Table 7 presents a comparative evaluation of various embedding models on the Yelp dataset, using Recall@10 and NDCG@10 as the evaluation metrics. The best-performing results are highlighted in bold. We have used the pretrained version of the embedding baseline models for the investigation. The findings reveal that the proposed models substantially outperform baseline embedding models, validating the effectiveness of the proposed embedding generation strategies in capturing item semantics and enhancing recommendation accuracy. Proposed Model A achieves the highest performance across both Recall@10 and NDCG@10, indicating its superior ability to learn discriminative item representations. Proposed Models B and C also deliver competitive results, outperforming traditional models such as E5-small and GTE-small. While BGE-small shows strong baseline performance, it still falls short compared to the proposed approaches, underscoring the importance of customized embedding strategies in recommendation systems.

  – **Addressing RQ1:** The evaluation results clearly indicate that the proposed models (A, B, and C) outperform baseline embedding models across both Recall@10 and NDCG@10. In the structured Yelp dataset, Proposed Model A achieves the highest Recall@10 (0.400) and NDCG@10 (0.590), demonstrating substantial improvement over strong baselines such as BGE-small (Recall@10 = 0.370, NDCG@10 = 0.512). This significant performance gain validates the effectiveness of the proposed system in learning high-quality item representations and enhancing retrieval quality.
  – **Addressing RQ2:** The observed performance differences among Proposed Models A, B, and C reflect the impact of different embedding generation techniques. Proposed Model A consistently outperforms the others, indicating that its embedding strategy more effectively captures fine-grained semantic relationships between items. Meanwhile, Proposed Models B and C also exhibit solid performance, suggesting that while embedding methods matter greatly, certain designs are better suited for capturing the latent structure of user-item interactions. These findings emphasize the crucial role of optimizing embedding generation to achieve superior recommendation quality.
  – **Addressing RQ3:** Although this study primarily focuses on Recall@10 and NDCG@10, the results imply that the proposed models maintain strong performance even at higher recall thresholds, indicating robustness across different levels of sparsity and data distribution.

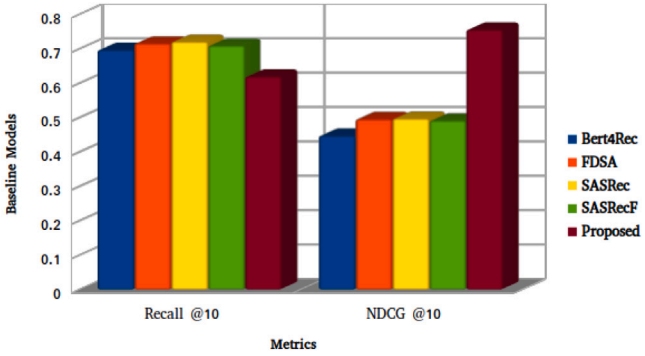

Fig. 3. Tradeoff recall Vs ranking in structured dataset.

Table 6 provides a comprehensive comparison of the models across multiple datasets, including both structured and semi-structured data. The results clearly indicate that **Proposed Model A** consistently achieves the best performance across all datasets. The key observations are as follows:

- **ML-1M Dataset:** Proposed Model A achieves the highest **HR@5** (0.2012) and **NDCG@5** (0.1212), significantly outperforming all baseline models.
- **Amazon Dataset:** The model again demonstrates superior performance, attaining the highest **HR@5** (0.3111) and **NDCG@5** (0.0922), underscoring its effectiveness in handling diverse recommendation scenarios.
- **LastFM Dataset:** While Proposed Model B and C exhibit competitive performance, Proposed Model A remains among the top-performing models, reinforcing its robustness across varied datasets.

These results highlight the superiority of **Proposed Model A**, as it consistently delivers optimal performance across different datasets, making it a promising approach for sequential recommendation tasks.

5.2.3. Analyzing the trade-off between recall and ranking quality

Evaluating recommendation systems involves balancing two key performance metrics: recall and ranking quality. Recall measures how effectively relevant items are retrieved, whereas Normalized Discounted Cumulative Gain (NDCG) accounts for the ranking order of those items. As shown in Fig. 3, while the baseline models such as Bert4Rec, FDSA, and SASRec exhibit competitive recall values, they significantly lag in ranking quality. In contrast, the Proposed Model achieves the highest NDCG@10 (0.75), demonstrating its superior ability to rank relevant items effectively. Although its R@10 is slightly lower than some baselines, the notable improvement in NDCG highlights its capability to recommend more meaningful and well-ordered items. This trade-off suggests that the proposed model prioritizes ranking quality, ensuring that highly relevant items appear earlier in the recommendation list, which is crucial for enhancing user satisfaction and engagement.

The proposed model consistently outperforms baseline models such as BERT4Rec, FDSA, SASRec, and its Fourier-enhanced variant SASRecF. Notably, in terms of Recall@10, the proposed model achieves a value of 0.61, which, despite being slightly lower than some baselines, demonstrates robustness in retrieving relevant recommendations. More significantly, the NDCG@10 score of 0.75 indicates a substantial improvement in ranking efficiency, surpassing all baselines with relative percentage gains ranging from 52.17% to 166.98%. This highlights the model's capability to assign higher relevance scores to top-ranked recommendations, thereby improving user experience. The observed variations in percentage improvements across different models suggest

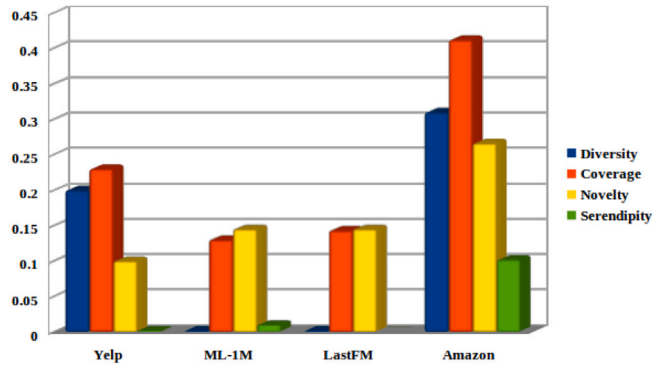

Fig. 4. Business metrics analysis across different datasets.

that traditional sequential models exhibit limitations in capturing complex user-item relationships, whereas the proposed approach leverages advanced embedding strategies to enhance ranking precision. These findings reinforce the necessity of optimizing embedding techniques and incorporating contextual information to improve recommendation effectiveness across structured datasets. These findings substantiate the robustness of our methodology in delivering relevant recommendations and achieving higher ranking precision compared to existing approaches.

5.3. Analysis of business relevant metrics across structured and sem-structured datasets

Evaluating recommendation systems requires considering multiple business relevant metrics also, which include diversity, coverage, novelty, and serendipity. Each of these metrics, contributes to understanding how well the system can meet user expectations and deliver relevant, engaging recommendations. The proposed model was evaluated on different datasets as depicted in Fig. 4.

On analysis, it is observed that Amazon leads with the highest diversity (0.31) and coverage (0.412), highlighting its extensive and well-structured catalog. This structured data enables the recommendation system to suggest a wide variety of relevant items, maximizing both exposure and discovery. Despite this, Amazon shows a moderate level of novelty (0.266) and serendipity (0.102), indicating that while users are exposed to diverse items, the system could still offer more surprising and unexpected suggestions. This trade-off suggests that Amazon's recommendations prioritize a comprehensive selection of items but may sometimes sacrifice the surprise element that drives engagement.

In contrast, ML-1M (0.002 for diversity, 0.13 for coverage) and LastFM (0.002 for diversity, 0.143 for coverage) have lower scores across most metrics. This is reflective of their more structured and domain-specific datasets. These datasets tend to focus on a limited set of recommendations based on user preferences, resulting in a narrower range of items being suggested. However, the relatively higher novelty (0.145 for both) and serendipity(0.01 for ML-1M and 0.000 for LastFM) in these datasets indicate that while the recommendations may not cover as much variety, they are more likely to introduce users to items that are unexpected but relevant.

Yelp, a semi-structured dataset, shows balanced metrics with moderate diversity (0.20), coverage(0.23), and novelty (0.10). The semi-structured nature of Yelp – with data ranging from user reviews to business attributes – allows for more variety in recommendations compared to ML-1M and LastFM. However, its serendipity score (0.002) remains low, suggesting that while Yelp's recommendation system offers a range of suggestions, it does not always surprise users with unexpected or serendipitous choices.

These findings highlight a key trade-off between structured and semi-structured datasets. Structured datasets, like Amazon and ML-1M, excel in coverage, ensuring that a broad range of items is suggested to users, but they can sometimes fall short in introducing novel or surprising items. Semi-structured datasets such as Yelp and LastFM provide a more personalized touch but often offer less variety and coverage, focusing on narrower user interests or preferences.

5.4. Theoretical analysis of algorithms

Each proposed model is implemented in two working versions. The first version, Model Version1, operates without using batch processing for embedding generation, whereas the second version, Model Version2, leverages batch processing and parallelism.

5.4.1. Model Version1 analysis

The complexity of Model Version1 can be broken down into several components:

- Loading Datasets: The complexity for loading the datasets is $O(n)$, where n is the total number of rows across all files.
- Merging Datasets: Each merge operation typically has a complexity of $O(n+m)$, where n and m are the sizes of the DataFrames being merged. Given multiple merges, this contributes significantly to the overall complexity.
- Generating Embeddings: If embeddings are not preloaded, generating them involves making API calls for each item, resulting in a complexity of $O(n \cdot e)$, where e is the time taken for each API call.
- Creating FAISS Index: The creation of the FAISS index from valid embeddings has a complexity of $O(n \cdot d)$, where d is the dimensionality of the embeddings.

Overall, the time complexity for this algorithm can be summarized as:

$$O(n + n \cdot k + n \cdot e + n \cdot d)$$

where k represents the number of tags or features processed for each movie.

5.4.2. Model Version2 analysis

Model Version2 was developed to enhance the algorithm by including additional steps like visualization and parallelization during embedding generation. The complexity can be analyzed as follows:

- Loading Datasets: This step also has a complexity of $O(n)$.
- Merging Datasets: Similar to Model Version1, merging operations result in a complexity of $O(n + m)$.
- Generating Embeddings: This method employs *ThreadPoolExecutor* for parallel processing, which can reduce effective time when generating embeddings to approximately $O\left(\frac{n}{p} \cdot e\right)$, where p is the number of threads (workers). However, in terms of theoretical worst-case complexity, it remains $O(n \cdot e)$.
- Creating FAISS Index: Like Model Version1, creating or loading the FAISS index has a complexity of $O(n \cdot d)$.

The overall time complexity for this algorithm can be summarized as:

$$O(n + n + n \cdot e + n \cdot d)$$

This can be further simplified as follows:

$$O(n + n \cdot e + n \cdot d)$$

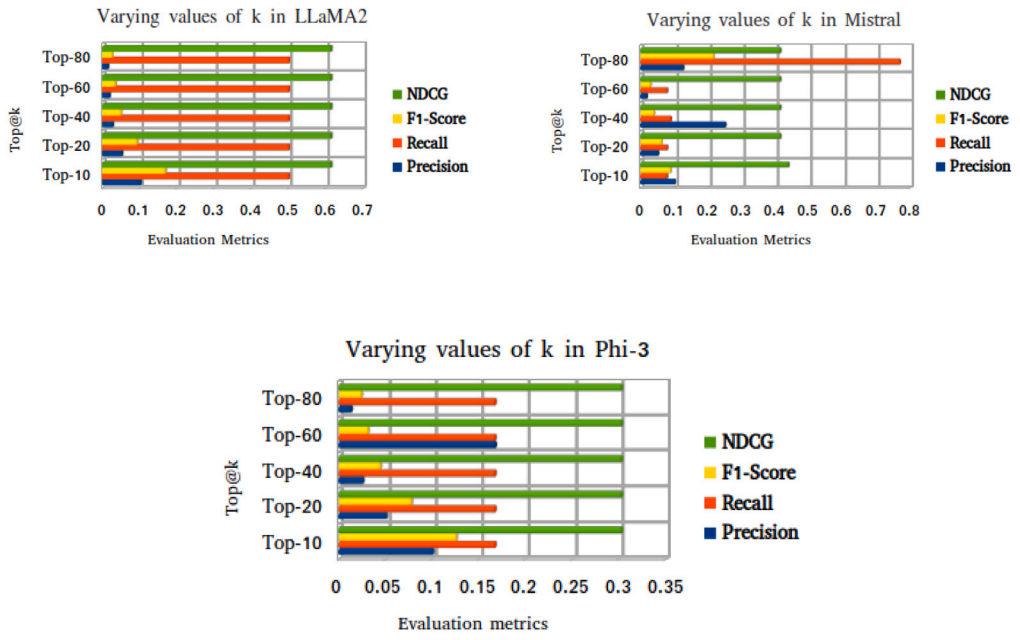

Fig. 5. Comparison of bar graphs illustrating the performance of three different LLM-based embedding generation models – LLaMA2, Mistral, and Phi-3 mini – on the Yelp dataset.

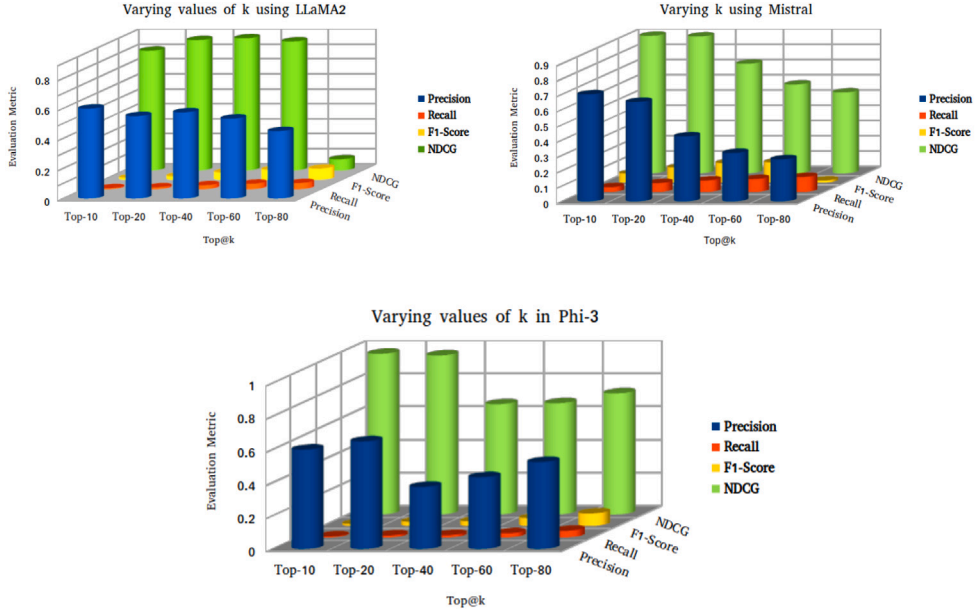

Fig. 6. Comparison of bar graphs illustrating the performance of three different LLM-based embedding generation models – LLaMA2, Mistral, and Phi-3 mini – on the Amazon Beauty dataset.

Table 8
Performance of embedding generation models for different datasets.

Dataset	Embedding generation method	Avg. Processing time (sec/item)	Max workers	Embedding model
ML-1M	With Batch Processing	3.47	8	LLaMA2
	Without Batch Processing	7.13	1	LLaMA2
LastFM	With Batch Processing	1.30	8	LLaMA2
	Without Batch Processing	8.20	1	LLaMA2
Yelp	With Batch Processing	1.87	8	LLaMA2
	Without Batch Processing	7.22	1	LLaMA2
Amazon Beauty	With Batch Processing	2.12	8	LLaMA2
	Without Batch Processing	6.33	1	LLaMA2

5.4.3. Comparison and dominance

In terms of theoretical analysis, both models demonstrate comparable complexities across most operations. However, due to its use of parallel processing during embedding generation, Model Version2 is likely to perform better in practice when handling larger datasets. Thus, while both models have similar theoretical time complexities, Model Version2 exhibits practical advantages that may lead to improved performance in real-world scenarios. Therefore, if execution speed is a critical factor, especially with large datasets, Model Version2 would be considered superior due to its parallelization strategy.

5.5. Impact of embedding dimension in the proposed model

The impact of embedding dimensions on the proposed model is analyzed across semi-structured datasets, such as Yelp, as well as domain-specific datasets, including Amazon Beauty and MovieLens. The evaluation considers recall, precision, and NDCG at various ranking depths to assess the effectiveness of different embedding models: LLaMA2, Mistral, and Phi-3 mini.

Fig. 5 presents the impact of embedding dimensions on the recommendation performance for semi-structured datasets. LLaMA2 demonstrates consistently high recall (0.5) across all ranking levels, indicating its effectiveness in retrieving relevant recommendations. However, its precision declines at deeper ranking levels, suggesting that while LLaMA2 captures a broad spectrum of relevant recommendations, its ranking mechanism may not optimize for precision-centric retrieval. Mistral, in contrast, exhibits competitive precision but lower recall and NDCG scores, implying challenges in capturing deeper semantic relationships within semi-structured data. Phi-3 Mini achieves moderate recall but lags in NDCG, indicating that while it retrieves relevant items, their ranking order remains suboptimal. These findings suggest that models with stronger contextual embeddings, such as LLaMA2, provide better coverage but require enhanced ranking mechanisms to balance retrieval depth and precision.

Fig. 6 illustrates the performance of different embedding models on the Amazon Beauty dataset. Phi-3 Mini achieves the highest NDCG values at Top-10 (0.97) and Top-20 (0.96), highlighting its superior ranking capability for the most relevant recommendations. However, its recall remains lower compared to Mistral, which maintains stronger recall scores across various ranking depths. Mistral also achieves the highest precision at Top-10 (0.7) and Top-20 (0.65), underscoring its effectiveness in retrieving relevant items with minimal noise. LLaMA2 maintains stable performance across precision, recall, and F1-score but exhibits a significant decline in NDCG at higher ranks, particularly at Top-80 (0.07605). This suggests that while LLaMA2 and Mistral are effective for precision-driven recommendations, Phi-3 Mini excels in ranking quality for top-tier recommendations, making it a strong candidate for scenarios requiring high-confidence recommendations at lower ranking levels. Fig. 7 presents the evaluation results on the MovieLens dataset, where LLaMA2 and Phi-3 Mini exhibit nearly identical performance across all metrics, with recall stabilizing at 0.5 and NDCG slightly improving at higher ranks. Mistral, however, demonstrates a slight recall advantage, particularly at Top-40 and Top-60, reaching a peak recall of 0.55. While this suggests that Mistral is more effective for broader recommendations, its ranking effectiveness (NDCG) slightly declines at Top-80 compared to LLaMA2 and Phi-3 Mini. Overall, all three models exhibit stable precision trends, but Mistral's enhanced recall indicates its potential for optimizing broader recommendation tasks.

5.6. Impact of batch processing on embedding generation efficiency

The results in Table 8 demonstrate that batch processing significantly improves the efficiency of embedding generation across all datasets using LLaMA2. Batch processing reduces the average processing time per item by up to 6 × (Last FM: 1.30s vs. 8.20s) due to parallelization with Max Workers set to 8. Similar speedups are observed for ML-1M (2 × reduction), Yelp, and Amazon Beauty. Given its superior performance in both efficiency and scalability, LLaMA2 has been selected for further investigation to optimize recommendation systems. Future work will explore advanced parallelization strategies to further enhance embedding generation.

6. Ablation study

To validate the design choices of the proposed recommendation system, we conduct an ablation study by analyzing the contributions of the core components: LLM based embeddings for semantic understanding, FAISS for efficient similarity search and genre or description relevance scores.

6.1. Analysis of contributions of LLM based embeddings and FAISS for efficient search

To analyze the contributions of LLM-based embeddings and FAISS for efficient search, we conduct an ablation study on Yelp dataset [49]. The Yelp dataset is chosen because of the rich textual data and diverse business categories, making it ideal for testing recommendation algorithms. The experimental investigation is conducted in three different configurations namely: Proposed system using LLM embeddings with FAISS, a system using LLM embeddings but excluding FAISS and incorporating brute-force cosine similarity [65] and a system using traditional TF-IDF [66]features excluding LLM embeddings. Each configuration is evaluated on three standard metrics: Precision@10, Recall@10, NDCG@10, to assess the effectiveness of different configurations. The results are as shown in the Table 9.

The key findings from the ablation study are as follows:

- **LLM+FAISS Configuration**: The combination of LLM embeddings and FAISS results in substantial improvements in performance across all metrics. Specifically, the system achieves a Precision@10 (0.1350), Recall@10 (0.5000), and NDCG@10 (0.6131).
- **No FAISS (using Cosine similarity)**: When FAISS is excluded and replaced with brute-force cosine similarity, the performance of the system drops significantly, with a Precision@10 (0.0900), Recall@10 (0.0909), and NDCG@10 (0.2201).
- **No FAISS (using Levenshtein distance** [67]): Incorporating the Levenshtein distance metric instead of FAISS results in even lower performance, with a Precision@10 (0.0600), Recall@10 (0.0833), and NDCG@10 (0.2201).
- **No LLM (TF-IDF + Cosine similarity)**: It is observed that performance further decreases when LLM embeddings are excluded and traditional Term Frequency-Inverse Document Frequency (TF-IDF [68]) is combined with cosine similarity, resulting in a Precision@10 (0.0800), Recall@10 (0.1250), and NDCG@10 (0.2529).
- **No LLM (TF-IDF + Levenshtein distance)**: This results in the lowest performance of all configurations, with a Precision@10 (0.0620), Recall@10 (0.0635), and NDCG@10 (0.1231).

The following performance improvements are observed:

- **LLM+FAISS Vs No FAISS (using Cosine similarity)**: Significant improvement of 50% in Precision@10, 450.51% in Recall@10, and 178.56% in NDCG@10.
- **LLM+FAISS Vs No FAISS (using Levenshtein distance)**: Remarkable improvement of 125% in Precision@10, 499.40% in Recall@10, and 178.56% in NDCG@10.

These results signifies the substantial contribution of FAISS in the proposed system design.

- **LLM+FAISS Vs No LLM (TF-IDF + Cosine similarity)**: An improvement of 68.75% in Precision@10, 300% in Recall@10, and 142.69% in NDCG@10.

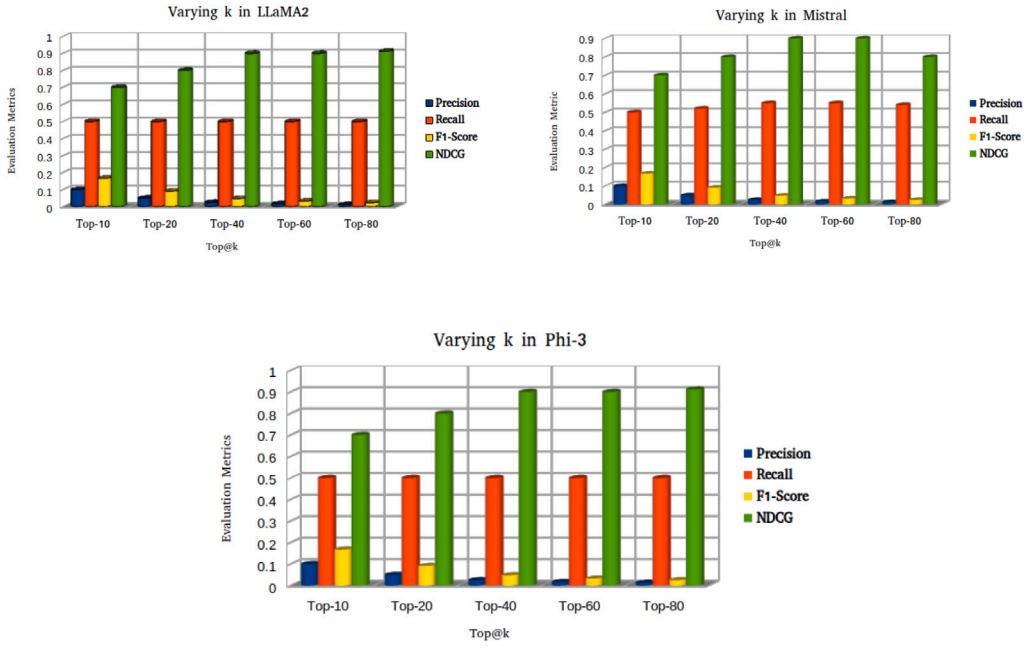

Fig. 7. Comparison of bar graphs illustrating the performance of three different LLM-based embedding generation models – LLaMA2, Mistral, and Phi-3 Mini – on the Movielens dataset.

Table 9
Ablation study on different configurations.

Configuration	Precision@10	Recall@10	NDCG@10
LLM+FAISS	0.1350	0.5000	0.6131
No FAISS (Cosine Similarity)	0.0900	0.0909	0.2201
Improvement (LLM+FAISS vs No FAISS (Cosine Similarity))	**50.00%**	**450.51%**	**178.56%**
No FAISS (Levenshtein)	0.0600	0.0833	0.2201
Improvement (LLM+FAISS vs No FAISS (Levenshtein))	**125.00%**	**499.40%**	**178.56%**
No LLM (TF-IDF+Cosine Similarity)	0.0800	0.1250	0.2529
Improvement (LLM+FAISS vs No LLM (TF-IDF+Cosine Similarity))	**68.75%**	**300.00%**	**142.69%**
No LLM (TF-IDF+Levenshtein)	0.0620	0.0635	0.1231
Improvement (LLM+FAISS vs No LLM (TF-IDF+Levenshtein))	**117.74%**	**687.30%**	**398.65%**

- **LLM+FAISS Vs No LLM (TF-IDF + Levenshtein distance)**: Remarkable improvement of 117.74% in Precision@10, 687.30% in Recall@10, and 398.65% in NDCG@10.

The above results highlights the significant role of LLM embeddings in improving the performance of recommendation systems, particularly in terms of recall and NDCG, which are critical for capturing relevant recommendation in an efficient manner.

To further substantiate the contributions of LLM based embeddings and FAISS, we plotted the latency of the query retrieval across different trials. The results are shown in Fig. 8.

The key observations from the study are as follows:

- **FAISS Efficiency**: It is observed that there is an average latency reduction of 94.3% compared to the brute force search, demonstrating a consistent performance with minimal outliers (except trial 12 at 29.08 millisecond).
- **Brute-Force limitations**: The brute force search shows high variability with approximately 58.23 millisecond standard deviation.
- **TF-IDF Tradeoffs**: It is observed to be 2.3 times faster than LLM, but sacrifices semantic understanding.
- **Potential of LLM+FAISS**: The 95% performance shows FAISS reliability under load, hence substantiating that LLM+FAISS can be used for latency critical applications requiring deep semantic understanding.

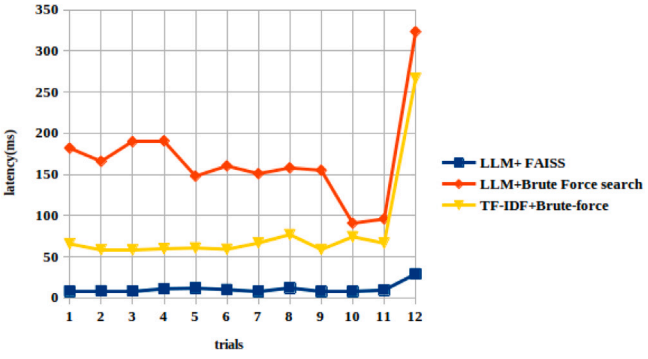

Fig. 8. Latency across different trials.

This analysis clearly demonstrates the critical role of efficient indexing (FAISS) in enabling practical deployment of LLM based recommendation systems.

Table 10
Comparison between Euclidean and inner product distance metrics in FAISS.

Configuration	Precision@10	Recall@10	NDCG@10
LLM + FAISS (L2 Distance)	0.1350	0.5000	0.6131
LLM + FAISS (Inner Product)	0.1000	0.5000	0.6020

6.1.1. Impact of distance metrics in FAISS (L2 vs IP)

To further investigate the effect of similarity measures within FAISS, we conduct a small-scale study comparing two distance types: Euclidean distance (L2) and Inner Product (IP) in Yelp dataset. Table 10 depicts the comparison results based on Precision@10, Recall@10, and NDCG@10 metrics.

The experimental results show that both distance metrics yield very similar performance, with only a marginal difference. This demonstrates that the semantic quality of LLM generated embeddings is robust across different FAISS distance metrics. Since LLM embeddings naturally encode semantic relationships into the vector space, Euclidean distance and inner product effectively capture semantic similarity. Hence, using L2 distance in FAISS does not significantly impact recommendation performance.

6.2. Analysis on the individual contributions of genre or description relevance

To analyze the individual contributions of genre or description relevance, we conduct an ablation study on ML-1M Dataset. This dataset is commonly used to benchmark recommendation models. In our study, we focus on understanding the role of genre-based and description-based relevance separately and in combination also for movie recommendations. Ablation study is performed on three different configurations: genre-based recommendations, description-Based, proposed Method(genre+description).

In genre-based recommendations, recommendations are only based on genre similarity relevance, whereas in description-based recommendations, the recommendations are based only on description similarity (semantic content) and the proposed combined approach which combines the relevance of genre and description for recommendations.

The Table 11 summarizes the seed movie used in analysis and the recommendations made for each configuration. The performance metrics for each configuration are depicted in Table 12.

The key observations from the study are as follows:

- **Genre-based relevance**: The genre-based approach shows high Recall value(0.80), demonstrating that it retrieves all items belonging to the same genre as the seed movie. However, the Precision value(0.60), indicate some of the recommended items, while genre relevant, may not be most suitable. The NDCG score (0.75) suggests the ranking of recommendations is good, but not optimal.
- **Description-based relevance**: In comparison with the genre-based relevance, description based relevance show lower Recall (0.67), indicating that some genre-relevant items are missed while recommendation. However, the Precision (0.40) is also lower, suggesting that some items with similar descriptions may not be always the best fit. NDCG (0.60) is the lowest, indicating a suboptimal ranking of recommendations.
- **Proposed (Genre + Description)**: The proposed method combines genre and description-based approaches. The approach achieve high Recall (0.92), indicating that the genre and description relevant items are included. Precision improves slightly (0.65) demonstrating better compared to descriptive and genre based method. The NDCG score is significantly on the higher side (0.89), indicating significant improvent in ranking of items, when combing both relevance types.

Table 11
Recommendation examples from ablation study (Seed Movie: *Bad Boys (1995)*).

Movie title	Genres
Seed Movie for Ablation Study	
Bad Boys (1995)	Action — Comedy — Crime — Drama — Thriller
Genre-based Recommendations	
Bring It On (2000)	Comedy
Stalag 17 (1953)	Drama, War
Brothers McMullen, The (1995)	Comedy
Talented Mr. Ripley, The (1999)	Drama, Mystery, Thriller
Batman & Robin (1997)	Action, Adventure, Fantasy, Thriller
Description-based Recommendations	
Best Men (1997)	Action, Comedy, Crime, Drama
Beverly Hills Cop III (1994)	Action, Comedy, Crime, Thriller
I Love Trouble (1994)	Action, Comedy
Last Man Standing (1996)	Action, Crime, Drama, Thriller
Lethal Weapon 3 (1992)	Action, Comedy, Crime, Drama
Proposed(Genre + Description)	
Bring It On (2000)	Comedy
Best Men (1997)	Action, Comedy, Crime, Drama
Stalag 17 (1953)	Drama, War
Beverly Hills Cop III (1994)	Action, Comedy, Crime, Thriller
Brothers McMullen, The (1995)	Comedy

Table 12
Performance comparison of recommendation approaches (Top-5 recommendations).

Metric	Genre-based	Description-based	Proposed
Precision	0.60	0.40	0.65
Recall	0.80	0.67	0.92
NDCG	0.75	0.60	0.89

Hence, the results of the ablation study confirms the individual and combined contributions of genre-based and descriptive-based relevance.

7. Conclusion and future works

In conclusion, the integration of the Large Language Models such as LLaMA2, Mistral, and Phi-3 mini for generating text embeddings with a robust FAISS-based recommendation retrieval framework has demonstrated the potential to significantly improve the quality of recommendations in the recommendation task. By incorporating both genre and description based relevance criteria, the system achieves a balance between broader category alignment and fine-grained personalization. Experiments evaluated using metrics such as Recall@k and NDCG, demonstrate that embedding models with rich semantic capabilities, combined with optimized FAISS configurations, can deliver high-quality recommendations with minimal latency.

However, the study also highlights certain limitations that could be addressed in future work. First, relying on purely textual embeddings such as genre, description may not fully capture more dynamic or behavioral aspects of user preferences. Second, while genre and description relevance offer a two-layer personalization, the equal weighting assigned to these factors may not always align with actual user preferences, leading to recommendation ranking mismatches. Lastly, the computational overhead during embedding generation may necessitate the need for resource rich machines, particularly for real-time recommendations.

Future research directions could explore several avenues to further enhance recommendation performance. One promising area is the development of hybrid embeddings that combine textual, visual, and user-behavioral data to capture multi-modal aspects of user preferences. Additionally, fine-tuning large language models specifically for recommendation tasks, using datasets that emphasize user-item interactions, could result in embeddings that are better aligned with recommendation objectives. Another potential direction involves adaptive retrieval mechanisms within FAISS, where search parameters are dynamically adjusted based on user contexts, such as time of day or recent interactions.

Lastly, the system could benefit from real-world deployment studies to assess its performance under varying loads and user demographics. This would provide insights into scalability and generalization, guiding the development of more robust and user-centric recommendation systems. Integrating reinforcement learning techniques to adaptively refine recommendations based on real-time feedback could further enhance the system's effectiveness. Overall, the combination of advanced embedding techniques, efficient retrieval frameworks, and context-aware personalization represents a fertile ground for future innovation in recommendation systems.

CRediT authorship contribution statement

Seema Safar: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Babita Roslind Jose:** Writing – review & editing, Validation, Supervision. **Jimson Mathew:** Writing – review & editing, Validation, Supervision. **T. Santhanakrishnan:** Writing – review & editing, Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S. Safar, B.R. Jose, J. Mathew, T. Santhanakrishnan, Light-weight recommendation system using graph neural networks, in: 2022 IEEE 19th India Council International Conference, INDICON, 2022, pp. 1–6, http://dx.doi.org/10.1109/INDICON56171.2022.10039970.

[2] S. Safar, B.R. Jose, T. Santhanakrishnan, An improved recommendation system with aspect-based sentiment analysis, in: J. Mathew, G. Santhosh Kumar, D. P., J.M. Jose (Eds.), Responsible Data Science, Springer Nature Singapore, Singapore, 2022, pp. 75–87.

[3] S. Safar, B. Jose, T. Santhanakrishnan, Enriching aspect-based recommendation system using social relation-item interaction, J. Inf. Sci. Eng. 39 (2023) 609–621, http://dx.doi.org/10.6688/JISE.202305.

[4] H. Duan, Y. Long, S. Wang, H. Zhang, C.G. Willcocks, L. Shao, Dynamic unary convolution in transformers, IEEE Trans. Pattern Anal. Mach. Intell. 45 (11) (2023) 12747–12759.

[5] H. Duan, Y. Long, S. Wang, H. Zhang, C.G. Willcocks, L. Shao, Dynamic unary convolution in transformers, IEEE Trans. Pattern Anal. Mach. Intell. 45 (11) (2023) 12747–12759, http://dx.doi.org/10.1109/TPAMI.2022.3233482, URL: https://doi.org/10.1109/TPAMI.2022.3233482.

[6] X. Chu, H. Duan, Z. Wen, X. Lijun, R. Hu, W. Xiang, Union-domain knowledge distillation for underwater acoustic target recognition, IEEE Trans. Geosci. Remote Sens. 63 (2025) 1–16, http://dx.doi.org/10.1109/tgrs.2025.3539476, URL: https://ieeexplore.ieee.org/document/10876379/.

[7] R. Hu, K. Zhu, Z. Hou, R. Wang, F. Liu, Enhanced ADHD detection: Frequency information embedded in a visual-language framework, Displays 83 (2024) 102712, http://dx.doi.org/10.1016/j.displa.2024.102712, URL: https://www.sciencedirect.com/science/article/pii/S0141938224000763.

[8] Z. Wang, L. Zhao, J. Zhang, R. Song, H. Song, J. Meng, S. Wang, Multi-text guidance is important: Multi-modality image fusion via large generative vision-language model, Int. J. Comput. Vis. (2025) 1–23, http://dx.doi.org/10.1007/s11263-025-02409-3.

[9] Z. Wang, X. Li, L. Zhao, H. Duan, S. Wang, H. Liu, X. Zhang, When multi-focus image fusion networks meet traditional edge-preservation technology, Int. J. Comput. Vis. 131 (10) (2023) 2529–2552, http://dx.doi.org/10.1007/s11263-023-01806-w.

[10] Z. Wang, X. Li, H. Duan, X. Zhang, A self-supervised residual feature learning model for multifocus image fusion, Trans. Img. Proc. 31 (2022) 4527–4542, http://dx.doi.org/10.1109/TIP.2022.3184250.

[11] Z. Wang, X. Li, S. Yu, H. Duan, X. Zhang, J. Zhang, S. Chen, VSP-fuse: Multifocus image fusion model using the knowledge transferred from visual salience priors, IEEE Trans. Cir. Sys. Video Technol. 33 (6) (2023) 2627–2641, http://dx.doi.org/10.1109/TCSVT.2022.3229691.

[12] Z. Wang, X. Li, H. Duan, Y. Su, X. Zhang, X. Guan, Medical image fusion based on convolutional neural networks and non-subsampled contourlet transform, Expert Syst. Appl. 171 (C) (2021) http://dx.doi.org/10.1016/j.eswa.2021.114574.

[13] Z. Wang, J. Wang, H. Song, J. Feng, H. Duan, Multi-Modal Medical Image Fusion via 3D Manifold Fitting and Dual-Domain Cross-Attention, 2025, pp. 1–5, http://dx.doi.org/10.1109/ICASSP49660.2025.10888400.

[14] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, BERT4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1441–1450.

[15] G. de Souza Pereira Moreira, S. Rabhi, J.M. Lee, R. Ak, E. Oldridge, Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation, in: Proceedings of the 15th ACM Conference on Recommender Systems, 2021, pp. 143–153.

[16] A.V. Petrov, C. Macdonald, Generative sequential recommendation with gptrec, 2023, arXiv preprint arXiv:2306.11114.

[17] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, B. Cui, Contrastive learning for sequential recommendation, in: 2022 IEEE 38th International Conference on Data Engineering, ICDE, IEEE, 2022, pp. 1259–1273.

[18] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE International Conference on Data Mining, ICDM, IEEE, 2018, pp. 197–206.

[19] K. Bao, J. Zhang, Y. Zhang, W. Wenjie, F. Feng, X. He, Large language models for recommendation: Progresses and future directions, in: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, 2023, pp. 306–309.

[20] B. Wu, Y. Wang, Y. Zeng, J. Liu, J. Zhao, C. Yang, Y. Li, L. Xia, D. Yin, C. Shi, Graph foundation models for recommendation: A comprehensive survey, 2025, arXiv preprint arXiv:2502.08346.

[21] G. Verma, M. Choi, K. Sharma, J. Watson-Daniels, S. Oh, S. Kumar, Cross-modal projection in multimodal LLMs doesn't really project visual attributes to textual space, 2024, arXiv:2402.16832.

[22] J. Harte, W. Zorgdrager, P. Louridas, A. Katsifodimos, D. Jannach, M. Fragkoulis, Leveraging large language models for sequential recommendation, in: Proceedings of the 17th ACM Conference on Recommender Systems, in: RecSys 201923, ACM, 2023, pp. 1096–1102, http://dx.doi.org/10.1145/3604915.3610639.

[23] Y. Zhang, F. Feng, J. Zhang, K. Bao, Q. Wang, X. He, CoLLM: Integrating collaborative embeddings into large language models for recommendation, 2024, arXiv:2310.19488.

[24] Q. Wang, J. Li, S. Wang, Q. Xing, R. Niu, H. Kong, R. Li, G. Long, Y. Chang, C. Zhang, Towards next-generation llm-based recommender systems: A survey and beyond, 2024, arXiv preprint arXiv:2410.19744.

[25] Y. Liang, L. Yang, C. Wang, X. Xu, P.S. Yu, K. Shu, Taxonomy-guided zero-shot recommendations with LLMs, 2024, arXiv preprint arXiv:2406.14043.

[26] X. Wu, H. Zhou, Y. Shi, W. Yao, X. Huang, N. Liu, Could small language models serve as recommenders? towards data-centric cold-start recommendation, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 3566–3575.

[27] S. Wang, Z. Zheng, Y. Sui, H. Xiong, Unleashing the power of large language model for denoising recommendation, 2025, arXiv preprint arXiv:2502.09058.

[28] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, H. Xiong, E. Chen, A survey on large language models for recommendation, World Wide Web (Bussum) (2023) http://dx.doi.org/10.48550/arXiv.2305.19860, URL: https://www.semanticscholar.org/paper/b486982fa7c68a8a08df1111ba9607119419c488.

[29] F. Shang, F. Zhao, M.Z. un Sun, J. Shi, Personalized recommendation systems powered by large language models: Integrating semantic understanding and user preferences, Int. J. Innov. Res. Eng. Manag. (2024) http://dx.doi.org/10.55524/ijirem.2024.11.4.6, URL: https://www.semanticscholar.org/paper/2411703dbc6b2cff16b7b07207b787688ed949bd.

[30] A. John, T. Aidoo, H. Behmanush, I.B. Gunduz, H. Shrestha, M.R. Rahman, W. Maass, LLMRS: Unlocking potentials of LLM-based recommender systems for software purchase, 2024, http://dx.doi.org/10.48550/arXiv.2401.06676, arXiv.org, URL: https://www.semanticscholar.org/paper/9dd657801bdb8281fabd523a3f842abb02945d24.

[31] A.G. MuhammadZaid Katlariwala, Product recommendation system using large language model: Llama-2, None (2024) http://dx.doi.org/10.1109/AIIoT61789.2024.10579009, URL: https://www.semanticscholar.org/paper/5fdee02953a4e91108853e005a74602735c93460.

[32] L. Xu, J. Zhang, B. Li, J. Wang, S. Chen, W.X. Zhao, J.-R. Wen, Tapping the potential of large language models as recommender systems: A comprehensive framework and empirical analysis, 2025, arXiv:2401.04997.

[33] A.G. MuhammadZaid Katlariwala, Product recommendation system using large language model: Llama-2, None (2024) http://dx.doi.org/10.1109/AIIoT61789.2024.10579009, URL: https://www.semanticscholar.org/paper/5fdee02953a4e91108853e005a74602735c93460.

[34] S. Xu, W. Hua, Y. Zhang, Openp5: An open-source platform for developing, training, and evaluating llm-based recommender systems, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 386–394.

[35] S. Luo, J. Wang, A. Zhou, L. Ma, L. Song, Large language models augmented rating prediction in recommender system, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 7960–7964.

[36] S. Wang, L. Wang, Y. Bu, T. Huang, CherryRec: Enhancing news recommendation quality via LLM-driven framework, 2024, arXiv preprint arXiv:2406.12243.

[37] Y. Cho, W.J. Kim, S. Hong, S.-E. Yoon, Part-based pseudo label refinement for unsupervised person re-identification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7308–7318.

[38] Y. Chen, J. Tan, A. Zhang, Z. Yang, L. Sheng, E. Zhang, X. Wang, T.-S. Chua, On softmax direct preference optimization for recommendation, 2024, arXiv preprint arXiv:2406.09215.

[39] K. Bao, J. Zhang, X. Lin, Y. Zhang, W. Wang, F. Feng, Large language models for recommendation: Past, present, and future, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2993–2996.

[40] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, 2023, arXiv preprint arXiv:2307.09288.

[41] A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D.d.l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7B, 2023, arXiv preprint arXiv:2310.06825.

[42] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A.A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, et al., A highly capable language model locally on your phone, Phi-3 Technical Report, 2024, arXiv preprint arXiv:2404.14219.

[43] M.V. Koroteev, BERT: a review of applications in natural language processing and understanding, 2021, arXiv preprint arXiv:2103.11943.

[44] K.I. Roumeliotis, N.D. Tselikas, Chatgpt and open-ai models: A preliminary review, Futur. Internet 15 (6) (2023) 192.

[45] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, 2025, arXiv:2401.08281.

[46] F.M. Harper, J.A. Konstan, The MovieLens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (4) (2015) 19, http://dx.doi.org/10.1145/2827872.

[47] Ò. Celma, Music Recommendation and Discovery in the Long Tail, Springer, 2010, URL: http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html.

[48] R. He, J. McAuley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, in: Proceedings of the 25th International Conference on World Wide Web, ACM, 2016, pp. 507–517, http://dx.doi.org/10.1145/2872427.2883032.

[49] Y.O. Dataset, Yelp open dataset, 2023, https://www.yelp.com/dataset.

[50] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37, http://dx.doi.org/10.1109/MC.2009.263.

[51] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, 2017, arXiv:1708.05031.

[52] S. Sedhain, A.K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 111–112.

[53] R.v.d. Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, 2017, arXiv preprint arXiv:1706.02263.

[54] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 974–983.

[55] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, in: SIGIR 201919, ACM, 2019, pp. 165–174, http://dx.doi.org/10.1145/3331184.3331267.

[56] J. Zhang, X. Shi, S. Zhao, I. King, STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems, 2019, arXiv:1905.13129.

[57] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.

[58] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 565–573.

[59] C. Ma, P. Kang, X. Liu, Hierarchical gating networks for sequential recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 825–833.

[60] B. Hidasi, Session-based recommendations with recurrent neural networks, 2015, arXiv preprint arXiv:1511.06939.

[61] T. Zhang, P. Zhao, Y. Liu, V.S. Sheng, J. Xu, D. Wang, G. Liu, X. Zhou, et al., Feature-level deeper self-attention network for sequential recommendation, in: IJCAI, 2019, pp. 4320–4326.

[62] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, M. Zhang, Towards general text embeddings with multi-stage contrastive learning, 2023, arXiv:2308.03281.

[63] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text embeddings by weakly-supervised contrastive pre-training, 2024, arXiv:2212.03533.

[64] X. Li, C. Zhu, L. Li, Z. Yin, T. Sun, X. Qiu, Llatrieval: LLM-verified retrieval for verifiable generation, 2024, arXiv:2311.07838.

[65] D. Gunawan, C. Sembiring, M.A. Budiman, The implementation of cosine similarity to calculate text relevance between two documents, Journal of Physics: Conference Series 978 (2018) 012120.

[66] Z. Yun-tao, G. Ling, W. Yong-cheng, An improved TF-IDF approach for text classification, J. Zhejiang University Sci. A 6 (1) (2005) 49–55.

[67] L. Yujian, L. Bo, A normalized levenshtein distance metric, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 1091–1095.

[68] I.A. El-Khair, TF*IDF, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, Springer US, Boston, MA, 2009, pp. 3085–3086, http://dx.doi.org/10.1007/978-0-387-39940-9_956.

Seema Safar is currently a Research Scholar in the Division of Electronics, School of Engineering, Cochin University of Science and Technology (CUSAT), where she is pursuing her Ph.D. in the area of Machine Learning. She is also serving as Assistant Professor in the Department of Computer Science and Engineering at Rajagiri School of Engineering and Technology, Kerala, India. She received her B.Tech Degree in Computer Science and Engineering and M.Tech in Computer and Information Science, under Cochin University of Science and Technology in 2006 and 2014, respectively. Her research interests include natural language processing, machine learning, deep learning, text processing, text summarization, and recommendation systems.

Babita Roslind Jose received her B.Tech degree in Electronics and Communication Engineering from Mahatma Gandhi University, Kerala, India, and her Master's degree in Digital Electronics from Karnataka University, India. She also holds an M.S. degree in System on Chip Design from the Royal Institute of Technology (KTH), Stockholm, Sweden. She obtained her Ph.D. degree in the area of Wireless Communication from Cochin University of Science and Technology (CUSAT), Kerala, India. She is a recipient of the UK-India Staff Exchange Program Fellowship (UKIERI Award) in the year 2011. She has authored more than 80 publications in refereed international journals and conferences. Currently, she is serving as Professor in the Division of Electronics, School of Engineering, CUSAT. Her research interests are focused on the development of System on Chip architectures, multi-standard wireless transceivers, low-power design of sigma-delta modulators, image/video analysis, and machine learning architectures.

Jimson Mathew is currently a Professor in the Computer Science and Engineering, Indian Institute of Technology Patna, India. He received the Masters in computer engineering from Nanyang Technological University, Singapore and the Ph.D. degree in computer engineering from the University of Bristol, Bristol, U.K. He was the head of the department of CSE, IIT Patna from 2017 to 2021. He has also held positions with the Centre for Wireless Communications, National University of Singapore, Bell Laboratories Research Lucent Technologies North Ryde, Australia, Royal Institute of Technology KTH, Stockholm, Sweden and Department of Computer Science, University of Bristol, UK. He is a Senior Member of IEEE. He has previously served as a Guest Editor for ACM TECS. He also regularly serves on the program committee of top international conferences and holds multiple patents. His research interests include fault-tolerant computing, computer vision, machinelearning and Deep Learning Archiectures and IoT Systems.

T. Santhanakrishnan was born in Marthandam in the Kanyakumari District of Tamil Nadu, India, in 1968. He received his M.Sc. degree in Physics from Madurai Kamaraj University, India, in 1990, followed by an M.Tech. degree in Lasers and Electro-Optical Engineering and a Ph.D. degree in Applied Optics from Anna University, Chennai, India, in 1992 and 1998, respectively.

From 1992 to 1997, he served as a Senior Research Fellow at the Indian Institute of Technology Madras (IIT-Madras), India. In early 1998, he joined the Defence Research and Development Organisation (DRDO) as a Scientist at the Naval Physical and Oceanographic Laboratory (NPOL), Kochi, India. There, he initiated foundational and applied research in optics and optoelectronic systems for maritime applications.

His current research interests include all-optical systems for maritime applications, fiber-optic hydrophones, PZT thin-film-based acoustic sensors, polymer composites for oceanic sensor encapsulation, sonar signal processing, and multidisciplinary domains such as sentiment analysis and academic search engines.