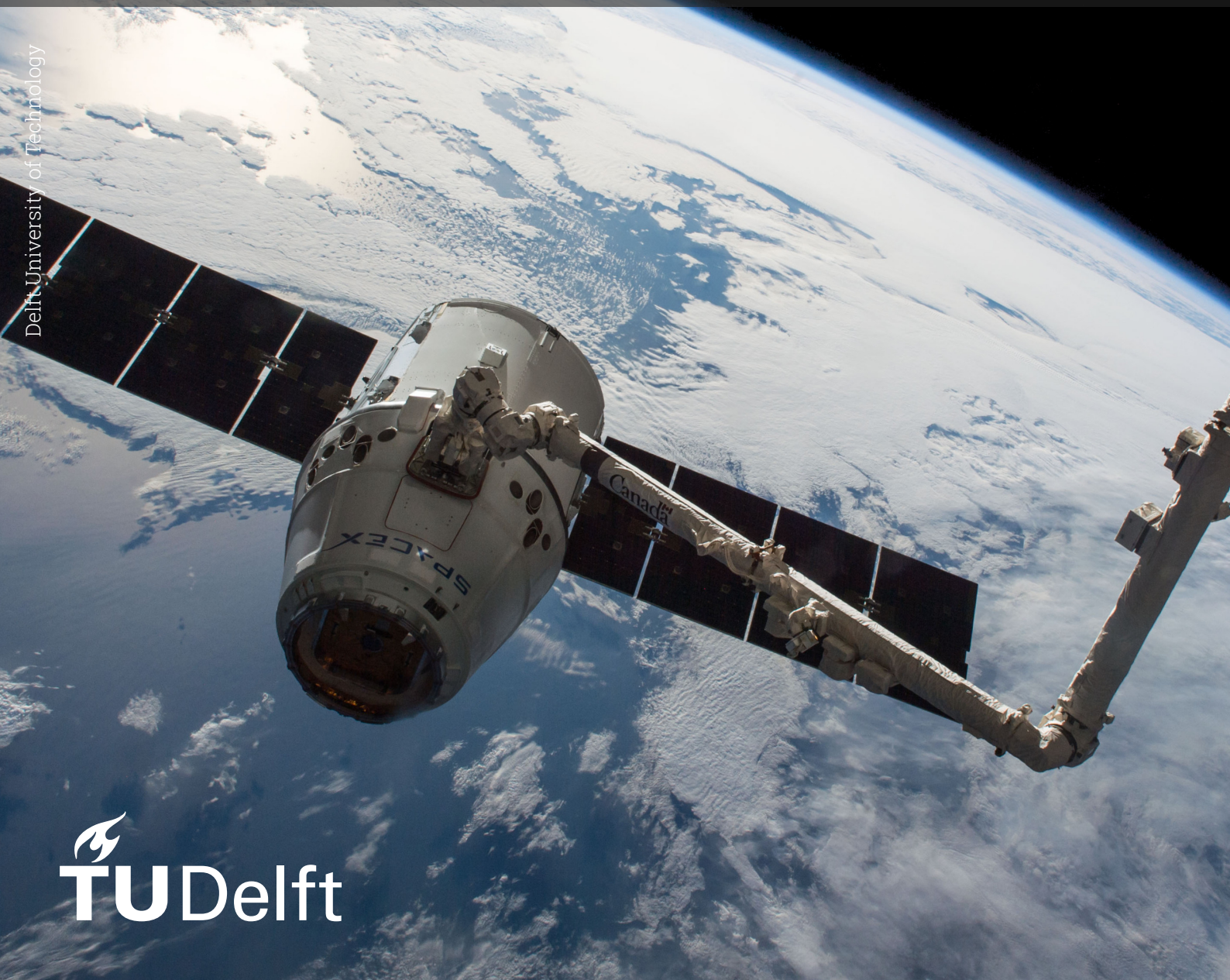
Comparative Analysis of Recommendation Models on Scopus Data

Unveiling Patterns in Sparse Interactions for Academic Discovery

Bugra Veysel Yildiz



Delft University of Technology

TU Delft

Comparative Analysis of Recommendation Models on Scopus Data

Unveiling Patterns in Sparse Interactions for Academic Discovery

by

Bugra Veysel Yildiz

TU Delft Supervisor:	Masoud Mansoury
Elsevier Supervisor:	Amin Tabatabaei
Scopus Product Owner:	Dilshad Begum Shaik
Project Duration:	November, 2024 - August, 2025
Faculty:	Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

This thesis marks the culmination of my Master's studies in Computer Science at TU Delft and reflects my research collaboration with Elsevier. Over the past year, I have had the opportunity to explore the design, development, and evaluation of a scalable academic article recommendation framework for Scopus. The project emerged from a shared ambition: to enhance literature discovery for millions of researchers by introducing intelligent, personalized, and context-aware recommendations.

Working on this research has been both challenging and rewarding. The scale and complexity of Scopus data, coupled with the need to address cold-start and sparse-interaction scenarios, demanded careful methodological choices and extensive experimentation. Beyond developing algorithms, the project allowed me to work at the intersection of academic research and industry application, balancing scientific rigor with practical deployment considerations.

I am deeply grateful to my TU Delft supervisor, **Masoud Mansoury**, for his guidance, encouragement, and invaluable feedback throughout this journey. I would also like to thank my Elsevier supervisor, **Amin Tabatabaei**, for his insights, support, and for providing the opportunity to work with real-world datasets and production-oriented challenges. I also extend my thanks to **Dan Li**, the former Data Lead of this project, for her support during the research period. Special thanks go to my colleagues and peers who shared ideas, offered advice, and made the research process more collaborative and enjoyable.

Finally, I would like to thank my family and friends for their unwavering support, patience, and belief in me throughout this demanding yet fulfilling period.

Bugra Veysel Yildiz
Delft, August 2025

Contents

Abstract

This thesis presents the design, implementation, and evaluation of a scalable, modular recommendation framework for academic article discovery on the Scopus platform. The research addresses limitations in Scopus's existing "Related documents" module, which produces static, non-personalized suggestions based solely on metadata keyword overlap. To overcome these constraints, the proposed framework introduces a dual-mode retrieval strategy capable of generating both personalized recommendations, informed by historical user interactions, and non-personalized recommendations, based solely on the context of a target article.

The study begins with an extensive exploratory data analysis (EDA) of 2024 Scopus interaction logs, comprising over 31 million user-item events. A novel data transformation pipeline is developed to convert implicit feedback signals, such as downloads, views, and exports, into continuous-valued preference scores that are suitable for collaborative filtering models. This enables the application of state-of-the-art algorithms despite the absence of explicit ratings.

Four recommendation models are implemented and compared: Bayesian Personalized Ranking (BPR), Factored Item Similarity Model (FISM), Light Graph Convolutional Network (LightGCN), and Knowledge Graph Attention Network (KGAT). Model evaluation is performed using both traditional offline ranking metrics (Recall@10, Precision@10, NDCG@10, MRR@10, Hit Rate@10) and a novel Large Language Model (LLM) based evaluation framework leveraging GPT-4o for semantic assessment of relevance and serendipity.

Results show that LightGCN consistently outperforms other models in both personalized and non-personalized scenarios, achieving the highest accuracy and scalability. Non-personalized recommendations remain valuable in cold-start and anonymous browsing contexts. The integration of LLM based evaluation offers deeper qualitative insights into recommendation quality, capturing semantic alignment and novelty beyond what is reflected in traditional metrics.

The proposed framework demonstrates that a unified embedding based architecture can effectively serve heterogeneous recommendation needs on large-scale scholarly platforms. The methodology and findings have broader implications for the design of academic recommender systems in data sparse and mixed user environments.

Keywords: Scopus, recommendation systems, collaborative filtering, LightGCN, implicit feedback, LLM based evaluation, academic discovery

1

Introduction

Scopus, developed by Elsevier, is among the most comprehensive abstract and citation databases for peer-reviewed literature. It indexes scholarly journals, books, and conference proceedings across a wide range of disciplines, including the life sciences, physical sciences, social sciences, and health sciences. As a central component of the global research infrastructure, Scopus supports academic discovery, citation tracking, impact assessment, and institutional benchmarking. With coverage exceeding 100 million records from over 25,000 titles, and with bibliometric indicators such as citation counts, h-indexes, and journal quartiles widely used in research evaluation and funding applications, Scopus plays a pivotal role in shaping how scientific knowledge is disseminated and consumed [20].

Despite the platform's scale and impact, the current implementation of content discovery mechanisms within Scopus remains limited in its effectiveness. The article detail pages feature a "Related documents" module, which produces suggestions based on surface-level metadata similarities, specifically keyword overlap across fields such as title, abstract, author names, and reference lists. These recommendations are static and identical for all users viewing the same document. They are not personalized, nor do they adapt to behavioral signals such as search history, article downloads, or reading patterns. Moreover, there is no mechanism for learning from user interactions to improve future suggestions. As a result, the current system exhibits low engagement levels, with click-through rates averaging approximately 2%, despite over two million visits per month to article pages.

To address these limitations, the 2024 Scopus product roadmap prioritizes the introduction of a document recommendation system capable of delivering relevant and context-aware content suggestions. The objective is to enhance user engagement, with a measurable goal of increasing click-through rates to a minimum of 5%, while also supporting broader product-level commercial strategies. The system must be capable of serving both identified users with interaction history and anonymous users lacking behavioral context.

This study addresses the underlying problem by designing and evaluating a modular recommendation framework for scholarly articles in the Scopus environment. Unlike traditional recommendation platforms that rely on explicit user-item ratings, Scopus does not collect such feedback, posing a significant modeling challenge [1]. To address this, the research introduces a data transformation approach that interprets implicit interaction signals, such as views, downloads, and exports as indicators of user interest. These signals were converted into continuous-valued preference scores through exploratory data analysis (EDA), resulting in a rating-like user-item matrix that supports collaborative filtering algorithms. This transformation process is one of the core contributions of this work and enables the application of recommender models in a context where no explicit feedback is available.

Using the interaction-derived dataset, this study develops and evaluates two distinct recommendation strategies: personalized and non-personalized. The personalized strategy generates user-specific recommendations by leveraging learned representations from historical user-item interactions, capturing individual behavioral preferences. In contrast, the non-personalized strategy produces recommendations that are solely dependent on a given target article, without requiring any user context or identity.

Figure 1.1: The current “Related documents” module on the Scopus article detail page. Recommendations are based solely on metadata keyword overlap and are not personalized or adapted to user behavior.

This dual-mode design departs from standard recommendation paradigms, which typically operate under the assumption that recommendations are always generated per user from the regular recommender system frameworks [31]. By explicitly supporting both user-aware and item-centric retrieval, the proposed framework accommodates a broader range of deployment scenarios, including cold-start conditions and anonymous browsing. This design is particularly novel in the academic domain, where recommendation contexts are often heterogeneous and user data availability is inconsistent. The comparative analysis of both strategies provides valuable insights into their respective performance trade-offs, offering a more flexible and adaptive solution for large-scale scholarly platforms such as Scopus.

To support these strategies, multiple state-of-the-art collaborative filtering models’ implementations have been used to fit the data of the Scopus for training and evaluated. These include Bayesian Personalized Ranking (BPR) [25], Factorized Item Similarity Models (FISM) [14], Light Graph Convolutional Networks (LightGCN) [11], and Knowledge Graph Attention Networks (KGAT) [30]. Each model offers different mechanisms for capturing latent relationships in user-item interaction graphs and is evaluated under varying conditions of data sparsity and content domain structure. All models are trained on the pseudo-rating matrix generated from the EDA process and optimized for top-K recommendation performance.

Evaluation in this study is carried out using a combination of quantitative and qualitative methods. To establish baseline performance, standard ranking metrics such as Recall@10 and NDCG@10 are used to measure the accuracy and effectiveness of the recommendation outputs in top-K retrieval settings. These metrics are widely adopted in the recommender systems literature and offer a reliable and reproducible means of comparing algorithmic performance across different recommendation strategies and model architectures [35].

However, in scholarly environments where user information needs are often exploratory, multifaceted, and context-dependent, these conventional metrics may fail to fully capture the perceived relevance or utility of a recommendation. Academic articles vary in purpose, contribution type, and methodological orientation, making it difficult to evaluate their relatedness through purely numeric measures. To address this limitation, large language models (LLMs) are incorporated as an additional evaluation mechanism. These models are used to assess semantic alignment, topical coherence, and the potential for serendipitous discovery between the target article and the recommended outputs. By interpreting content-level relationships beyond what is reflected in interaction data, LLMs provide a deeper and more interpretable layer of evaluation that complements traditional metrics, especially in domains with sparse or noisy user feedback.

This work contributes a modular recommendation framework for academic discovery systems that is scalable, interpretable, and adaptable to real-world deployment constraints. Its key contributions include a data transformation pipeline for implicit signal processing, a dual-mode recommendation strategy designed for hybrid user scenarios, and a comparative evaluation of collaborative filtering models within a scholarly platform context. Together, these contributions support the advancement of article recommendation within Scopus and inform the design of recommendation systems in broader academic information retrieval environments.

The remainder of this document is organized as follows. Chapter 2 reviews related work on academic recommendation systems, covering collaborative filtering approaches, graph-based models, content-based methods, and LLM-based evaluation. Chapter 3 presents the exploratory data analysis (EDA) of the Scopus dataset, including data source description, preprocessing, interaction mapping, and sample construction for experiments. Chapter 4 outlines the research design, dual-mode recommendation generation strategy, implemented models, and evaluation methods. Chapter 5 provides a comparative analysis of experimental results across datasets and models, highlighting key performance insights. Chapter 6 discusses the limitations of the study in terms of data, infrastructure, and computational constraints. Finally, Chapter 7 concludes the thesis with a summary of findings and suggestions for future research.

2

Related Works

Academic recommendation systems have become increasingly important as the volume of scholarly publications continues to grow at an unprecedented rate. Unlike general-purpose recommenders, which often operate in domains with abundant user feedback and well-defined item attributes, academic recommenders must address domain-specific challenges such as specialized content, highly sparse interaction patterns, and complex relationships between papers, authors, and research domains. Prior surveys, such as that by Beel et al. [4], have shown that many research paper recommenders struggle with the cold-start problem and data sparsity, both of which limit their effectiveness. For large-scale scholarly platforms like Scopus, which index more than 100 million articles, effective recommendation systems are essential for enabling researchers to move beyond static keyword searches and discover relevant, high-impact content, as emphasized by Bai et al. [2].

This chapter reviews key strands of related work that inform the design and evaluation of the proposed framework. Section 2.1 examines collaborative filtering methods, including traditional matrix factorization techniques and models tailored for implicit feedback. Section 2.2 explores graph-based collaborative filtering approaches and their role in addressing sparse interaction data. Section 2.3 discusses content-based methods that leverage article metadata and textual representations to improve relevance, particularly in cold-start scenarios. Finally, Section 2.4 considers emerging work on large language model (LLM)-based evaluation, which offers a semantic perspective on recommendation quality beyond traditional ranking metrics.

2.1. Collaborative Filtering Approaches

Collaborative filtering techniques, particularly model-based approaches, have shown promising results in academic recommendation contexts. Koren introduced SVD++, which extends traditional Singular Value Decomposition by incorporating implicit feedback alongside explicit ratings [16]. This approach is particularly valuable in academic contexts where explicit ratings are rare but implicit signals (views, downloads, citations) are abundant. Koren's experiments demonstrated that SVD++ achieves significantly better Root Mean Squared Error (RMSE) than earlier models, with improvements of approximately 5% over basic SVD models. The algorithm's ability to combine explicit factors with implicit feedback vectors enables more accurate predictions even for users with limited explicit feedback, making it well-suited for academic recommendation scenarios where interaction data is typically sparse.

Bayesian Personalized Ranking (BPR), developed by Rendle et al., offers another powerful approach specifically designed for implicit feedback scenarios [26]. By modeling pairwise preferences rather than absolute ratings, BPR aligns well with academic recommendation contexts where interactions are typically binary. The algorithm builds user-specific rankings by maximizing the posterior probability of correct item ranking through stochastic gradient descent with bootstrap sampling. However, it has been identified that significant limitations in BPR's handling of the cold-start problem, noting that its performance declines by up to 30% for new users or items which is a common scenario in rapidly evolving research fields where new papers are constantly being published [10].

Addressing the data sparsity challenge prevalent in academic recommendation, Factored Item Similarity Models (FISM) model has been proposed [14]. This approach represents the item-item similarity matrix as the product of two low-dimensional latent factor matrices, enabling the capture of transitive relationships between items even when they have not been co-rated by any users. Their extensive experiments across multiple datasets showed that FISM consistently outperforms competing models like SLIM, ItemKNN, and BPRMF, with up to a 24% improvement in hit rate in the sparsest dataset configurations. This makes FISM particularly promising for academic recommendation, where user-item interactions are typically sparse and unevenly distributed across research domains.

2.2. Graph-Based Collaborative Filtering and Sparse Data Challenges

Graph-based collaborative filtering has emerged as a promising solution for recommendation in sparse settings, particularly through the development of models like LightGCN. Introduced by He et al. [11], LightGCN simplifies the conventional graph convolutional architecture by removing nonlinear transformations and feature projection layers. Instead, it focuses solely on neighborhood aggregation through linear embeddings, which significantly reduces computational overhead while preserving collaborative signal propagation. Their experiments on benchmark datasets (e.g., Amazon, Yelp) showed that LightGCN consistently outperforms traditional GCN-based models and classic matrix factorization approaches, particularly in scenarios where user-item interaction data is limited. These findings motivated its adoption in our study, where data sparsity and scale were prominent challenges.

In the academic domain, few studies have applied LightGCN directly to scholarly data due to the lack of large-scale, standardized benchmarks and the complexities involved in modeling heterogeneous relations (e.g., citations, co-authorship). However, Wu et al. [33] applied LightGCN to the scholarly platform AMiner and observed substantial improvements in hit rate and NDCG over BPR and NeuMF baselines when evaluating paper recommendations based on citation interactions. Their work demonstrated that LightGCN can effectively model implicit preferences between users and documents even when explicit feedback is absent.

Further, research by Liu et al. [19] on sparse interaction datasets in digital libraries found that LightGCN's multi-hop aggregation mechanism allows it to better capture transitive relationships among papers, which is critical in scientific domains where direct co-reading behavior is rare. In their experiments, LightGCN achieved up to a 15% gain in Recall@10 over FISM and MF on cold-start user splits, highlighting its strength in sparse academic contexts.

Studies outside academia have also explored the robustness of LightGCN in sparse regimes. For instance, Sun et al. [29] benchmarked LightGCN on industrial-scale recommendation datasets with over 1 billion interactions and showed its scalability through distributed training techniques, such as neighbor caching and subgraph sampling. Their findings emphasize that LightGCN retains its effectiveness even as user density drops, provided the graph is sufficiently connected through indirect associations. This supports our decision to use LightGCN on the year-long Scopus interaction dataset, where over 22 million items created a highly sparse interaction matrix.

In parallel, alternative research has tackled data sparsity from a hybrid modeling perspective. Wang et al. [30] proposed KGAT to integrate knowledge graphs into collaborative filtering, leveraging relation-aware attention mechanisms. While KGAT improves semantic modeling by aggregating neighborhood signals from citation and entity graphs, it introduces additional overhead, which may not be suitable for real-time or large-scale scenarios without specialized infrastructure.

These collective efforts reveal that while data sparsity remains a core limitation in academic recommendation systems, graph-based models such as LightGCN offer scalable and effective strategies for capturing collaborative signals when direct user-item feedback is limited. However, performance is still contingent on dataset connectivity, user engagement depth, and graph construction quality, all of which remain critical areas for continued research and optimization.

2.3. Content-Based Approaches

Content-based recommendation methods have become indispensable in academic recommendation systems, particularly in mitigating the cold-start and data sparsity challenges that arise when user-item

interactions are minimal or highly skewed. Unlike collaborative filtering techniques, which depend on user behavior patterns, content-based models leverage intrinsic attributes of articles such as abstracts, titles, keywords, journal information, and author affiliations to infer relevance and similarity. These approaches are especially important in academic domains where new publications emerge continuously and where explicit user ratings are rarely available.

One influential contribution is the Sparse Linear Method (SLIM) by Ning and Karypis [21], which learns a sparse item-item similarity matrix from interaction data using Lasso-based optimization. Although originally classified under collaborative filtering, SLIM's design naturally supports integration with metadata and content features, enhancing interpretability and scalability. SLIM has shown superior performance in top-N recommendation tasks compared to baseline models such as ItemKNN, WRMF, and PureSVD, particularly in datasets with high sparsity levels.

Recent advances have explored hybrid architectures that tightly integrate metadata into the learning process. For instance, the Content-Enhanced Collaborative Filtering (CECF) model proposed by Zhang et al. [36] combines user-item interaction matrices with content embeddings derived from abstracts and keyword vectors. Their results demonstrated consistent gains in precision and NDCG metrics across CiteULike and BibSonomy datasets, highlighting the value of semantic enrichment in sparse academic domains.

Another significant work is SPECTER by Cohan et al. [5], a Transformer-based document embedding model trained using citation relationships. SPECTER learns contextual embeddings that capture scholarly intent and semantic similarity, leading to substantial improvements in both related-article retrieval and downstream recommendation tasks. Compared to TF-IDF, doc2vec, and even some deep content models, SPECTER achieved over 30% better recall in several benchmark settings, underscoring the power of citation-informed semantic representations.

In a similar vein, BERTopicRec [3] applied topic modeling with BERT-based embeddings to enhance academic recommendations. By grouping articles based on thematic coherence and integrating these clusters into the recommendation engine, the system improved diversity and user satisfaction. Notably, this approach proved useful in early-stage recommendation pipelines where user history is sparse or unavailable.

Furthermore, DeepCF-HT, a heterogeneous model by Wang et al. [32], incorporates not only content metadata such as topic labels and journal fields but also user attributes into a deep neural collaborative filtering architecture. Their experiments showed that side information can contribute up to a 22% improvement in Recall@10, especially in cold-start and low-activity scenarios, conditions that mirror the sparsity observed in Scopus interaction data.

Additionally, Collins et al. [6] conducted a large-scale empirical study comparing various types of document-level representations for research paper recommendation. Their evaluation found that semantically rich embeddings generated from abstracts and keyphrases outperform traditional frequency-based methods by 15–20% in precision metrics. These findings emphasize that metadata quality, and not just its presence, plays a critical role in driving effective academic recommendations.

Despite their advantages, content-based approaches are not without limitations. Purely content-driven models often struggle to generalize user intent across disciplines or to recommend novel items beyond previously consumed topics. Nevertheless, as shown by Okura et al. [22], combining sequential behavior modeling with content representations significantly enhances personalization, especially in dynamic scholarly domains.

2.4. LLM-Based Evaluation Using Item Metadata in Recommendation Systems

The emergence of Large Language Models (LLMs), including GPT 3.5 and GPT 4, has introduced new possibilities for evaluating recommender systems that extend beyond the limitations of traditional numerical metrics. In particular, item-level metadata such as titles, abstracts, keywords, and domain-specific descriptors provides a rich contextual foundation that LLMs can interpret to assess the semantic coherence, topical relevance, and novelty of recommended items. This capability is especially valuable

in academic and scientific domains where user interaction signals are often limited and preferences are closely linked to nuanced content and subject-specific interests.

Recent studies have proposed LLMs as semantic evaluators that can contextualize recommendations through language-based reasoning. Sun et al. [28] introduced ChatEval, a framework in which ChatGPT is prompted with a user profile that is constructed from previously viewed item metadata along with a candidate article's abstract or title. The model is asked to evaluate the relevance and novelty of each recommendation using structured response formats. Their experimental findings on academic datasets demonstrate that scores generated by LLMs strongly align with human judgments of relevance and reveal subtle content alignment signals that are often overlooked by standard evaluation metrics such as Recall or NDCG. In addition, this framework yields natural language justifications, thereby enhancing interpretability and supporting the development of explainable recommendation systems.

From a broader methodological standpoint, Ji et al. [12] presented a comprehensive review of LLM applications in recommendation tasks, including their role in evaluating system outputs. The authors provide a classification framework for LLM-based evaluation use cases and highlight the importance of carefully designed prompts, particularly when working with structured textual metadata such as abstracts or topical descriptors. Their findings emphasize that LLMs are capable of uncovering latent semantic relationships between user interests and content features, offering a more refined understanding of recommendation quality in cases involving sparse feedback or highly specialized domains.

Integrating LLMs into the evaluation process marks a shift in how recommendation performance is assessed. Rather than relying solely on behavioral indicators such as clicks or ratings, LLMs enable evaluation through human-like reasoning over textual content. This approach is especially useful in academic settings where user feedback is limited, manual annotation is resource-intensive, and relevance often depends on individual intent, disciplinary context, and scholarly framing. LLM-based evaluation methods can also address complex dimensions such as novelty, cross-domain relevance, and the discovery of unexpected but useful content, which are difficult to quantify through conventional ranking metrics.

Nevertheless, there are challenges associated with LLM-based evaluation. One significant concern is the sensitivity to prompt design, as even minor changes in phrasing may lead to substantial differences in model output. Additionally, the computational demands can become significant when evaluating large-scale recommendation datasets, particularly when relying on commercial interfaces or high-capacity local deployments. Another issue is reproducibility, since the output of LLMs can vary across model versions or contexts unless strict control conditions are implemented. These limitations point to the need for rigorous prompt engineering, response calibration, and reproducibility protocols when incorporating LLMs into recommendation evaluation pipelines. Despite these challenges, the ability of LLMs to perform structured reasoning over item metadata provides a promising direction for developing semantically grounded and user-centric evaluation frameworks in recommendation system research.

3

Exploratory Data Analysis (EDA)

This section explains the exploratory data analysis (EDA) performed on the Scopus interaction dataset, which underpins the development of academic article recommendation models in this study. The primary aim of this analysis is to understand the underlying structure, distribution, and behavioral patterns present in the data, thereby informing the modeling pipeline, feature selection, and evaluation methodology. In the context of recommender systems, especially within academic domains, EDA plays a critical role in uncovering challenges such as data sparsity, user engagement variability, and interaction noise, all of which directly affect model performance and generalizability.

The dataset used in this project was derived from Adobe Analytics logs collected from the Scopus platform over the course of 2024. It captures rich implicit feedback signals from users engaging with academic content, including actions such as article downloads, full-text views, and the addition of articles to alert lists. Unlike mainstream recommendation platforms that rely heavily on explicit user ratings, the Scopus platform records only behavioral events, which must be carefully interpreted and transformed into usable signals for collaborative filtering and other recommendation algorithms.

A key novelty of this analysis lies in its role as a data transformation bridge between real-world academic interaction data and the input requirements of modern recommender system frameworks. Unlike typical domains where recommender systems are deployed—such as e-commerce, streaming services, or social media—Scopus does not collect or store explicit ratings or clearly quantifiable user preferences. Platforms like Netflix or Amazon, for example, offer numerical ratings or thumbs-up/down signals that directly map to user preferences [27]. In contrast, Scopus only logs passive interaction signals such as views, downloads, and exports, which require contextual interpretation. Due to this structural difference, the raw Scopus interaction data could not be used directly within widely adopted recommendation frameworks such as RecBole [37], which expect well-defined user-item rating matrices. Therefore, a dedicated preprocessing pipeline was developed in this EDA phase to transform these implicit behaviors into a pseudo-rating format. This step involved mapping each type of interaction to a preference intensity and generating a model-compatible user-item matrix. Without this foundational transformation, it would not have been possible to train or evaluate collaborative filtering models on Scopus data. This preprocessing not only enables compatibility with existing algorithms but also contributes a novel methodological approach for adapting sparse, implicit academic usage data to large-scale recommender system infrastructures.

A key novelty of this analysis lies in its role as a data transformation bridge between real-world academic interaction data and the input requirements of modern recommender system frameworks. Unlike typical domains where recommender systems are deployed such as e-commerce, streaming services, or social media, Scopus does not collect or store explicit ratings or clearly quantifiable user preferences. Platforms like Netflix or Amazon, for example, offer numerical ratings or thumbs-up/down signals that directly map to user preferences. In contrast, Scopus only logs passive interaction signals such as views, downloads, and exports, which require contextual interpretation. Due to this structural difference, the raw Scopus interaction data could not be used directly within widely adopted recommendation frame-

works such as RecBole, which expect well-defined user-item rating matrices [37]. Therefore, a dedicated preprocessing pipeline was developed in this EDA phase to transform these implicit behaviors into a pseudo-rating format. This step involved mapping each type of interaction to a preference intensity and generating a model-compatible user-item matrix. Without this foundational transformation, it would not have been possible to train or evaluate collaborative filtering models on Scopus data. This preprocessing not only enables compatibility with existing algorithms but also contributes a novel methodological approach for adapting sparse, implicit academic usage data to large-scale recommender system infrastructures.

The complete dataset contains a large number of users and interactions, along with over **100 million** academic articles. The resulting user-item matrix is highly sparse, which is a common characteristic in academic domains, making it important to understand both global and local distribution patterns to avoid biased or unstable model behavior. This sparsity necessitates strategic sampling, filtering, and preprocessing choices to ensure that the recommendation models are trained and evaluated under conditions that reflect real-world usage while remaining computationally feasible.

To address the challenges associated with large-scale data and to facilitate iterative experimentation, we constructed multiple representative samples of the full dataset. These samples vary in terms of user and item activity thresholds, temporal coverage, and interaction density. By doing so, we aim to simulate various academic use cases, such as supporting interdisciplinary discovery and addressing cold-start scenarios.

The analyses presented in the following subsections examine key structural properties of the dataset, including the distribution of user activity and item popularity, sparsity levels, interaction types, and time-based engagement trends. These insights serve as a foundation for informed algorithm selection, performance benchmarking, and the design of evaluation protocols tailored to the academic research context. Ultimately, this EDA sets the empirical groundwork for building robust, collaborative filtering and knowledge-aware article recommender systems on the Scopus platform.

3.1. Dataset Overview

3.1.1. Data Source and Collection

The dataset employed in this study originates from Adobe Analytics (AA) event logs systematically collected across the Scopus platform throughout the calendar year 2024. These logs constitute a comprehensive and fine-grained repository of user activity, capturing a broad spectrum of behavioral signals and system interactions associated with academic content discovery, exploration, and consumption. This extensive behavioral dataset provides an ideal foundation for constructing implicit-feedback-based recommendation models specifically tailored for scholarly information retrieval.

In its raw form, the dataset comprises approximately **57,743,094** event-level records, encompassing more than **1,000** unique variables. These variables span diverse categories, including session identifiers, user metadata, query logs, document-level attributes, interaction events, and platform navigation traces. The breadth and granularity of this dataset present both opportunities and challenges: while it enables sophisticated modeling of user behavior and contextual dynamics, its high dimensionality and variability necessitate rigorous preprocessing and feature selection to ensure relevance and tractability for downstream tasks.

Variable Selection Framework. To ensure the construction of a reliable and behaviorally informative user-item interaction dataset, we implemented a rigorous multi-stage variable selection framework that integrates both qualitative expert knowledge and quantitative evaluation techniques. This process was designed to filter the original high-dimensional feature space into a semantically meaningful and statistically robust subset of variables for downstream modeling.

The first stage, **Semantic Relevance**, involved mapping each variable to its theoretical utility in modeling user engagement, using internal documentation such as Scopus analytics manuals and Mendeley behavioral reports. This step ensured alignment between available features and the specific objectives of recommender system design.

Next, we assessed **Statistical Validity** by measuring the non-null coverage of each variable across the

dataset. Features with excessive missingness, defined as greater than 80% null values, were flagged for exclusion unless they offered unique and irreplaceable behavioral insight. This criterion ensured that retained variables would be both computationally viable and generalizable across the user base.

Finally, we evaluated **Behavioral Informativeness** by examining whether each feature captured meaningful and interpretable forms of user interaction. Variables directly tied to active content engagement, such as viewing abstracts or downloading full-text PDFs, were prioritized. In contrast, variables representing passive navigation events or incidental clicks were systematically excluded.

3.1.2. Variable Inventory Consolidation and Final Selection

In the development of our behavior-based recommendation framework, the selection of variables from Adobe Analytics (AA) logs and internal documentation (specifically the Mendeley Report) was foundational. The raw AA logs capture a vast array of user interaction data, often exceeding one thousand distinct parameters, many of which are redundant, sparsely populated, or ambiguously defined. To ensure modeling tractability and interpretability, we undertook a systematic consolidation process that integrated three sources: (i) the official Mendeley variable definitions, (ii) our internal AA tracking variable audit, and (iii) exploratory variables we identified from domain-specific observations.

The final version of the schema was categorized into three domains: session and user-related variables, query and document-related variables, and interaction-related variables. Each feature was evaluated based on its semantic clarity, completeness, behavioral relevance, and technical feasibility, such as data type, cardinality, and presence across sessions. Optional fields were recorded for exploratory or future use but excluded from core model training to maintain pipeline efficiency and reduce noise.

Focus on Interaction-Level Features

While variables related to sessions and queries provide essential context, our modeling emphasis centered on high-signal user interaction features that are most directly aligned with content consumption, interest expression, and downstream engagement. This focus is particularly important in academic environments where explicit feedback, such as ratings, is not available, and engagement must be inferred from actions such as viewing abstracts, saving documents, or downloading full-text files.

From the broader AA inventory, we selected a focused set of interaction-specific variables based on two primary criteria: (1) semantic alignment with meaningful user actions (as defined by Mendeley product and UX teams), and (2) sufficient coverage across user sessions to ensure reliable learning.

The final set of selected AA interaction variables includes:

- `prop3` (**action_taken**): This is the central behavioral feature used to distinguish various key user actions. Specific values retained include *add to my list*, *save alert*, *file downloads*, *content export*, and *content view*. These are considered high-confidence proxies for academic interest and were used to construct positive interaction labels in our recommendation models.
- `post_evar119` (**button_link**): This variable provides detailed insight into which document access elements the user engaged with. Selected values include *Full Document Link*, *download*, *full-text*, *citations*, *title*, *author*, *view-abstract*, *related documents*, and *view-at-publisher*. These UI elements are often directly linked to subsequent high-engagement behaviors.
- `post_evar124` (**button_type**): This field was specifically retained to capture abstract view interactions. The value *menu:show abstract* is used to identify cases where users opened detailed views, which are indicative of deeper interest in the article content.

Together, these variables formed the basis for our implicit feedback signal generation. Unlike passive telemetry, these interactions are considered intentional and purpose-driven, especially within a research-focused platform like Scopus. Consequently, they serve not only as features for collaborative filtering but also as ground truth signals for evaluating engagement, relevance, and serendipity in our downstream modeling and dashboard metrics. The final set of interaction-focused variables selected for modeling user engagement is summarized in Table 3.1.

Table 3.1: Selected Adobe Analytics Variables for Interaction Modeling

AA Variable Name	Column Name	Selected Values
prop3	action_taken	add to my list, save alert, file downloads, content export, content view
post_evar119	button_link	Full Document Link, download, full-text, citations, title, author, view-abstract, related documents, view-at-publisher
post_evar124	button_type	menu:show abstract

3.2. Interaction Mapping

A key component of the data preprocessing pipeline involved translating the wide range of user behaviors captured in the Scopus platform into a standardized set of interaction scores. Unlike platforms that collect explicit feedback such as ratings or thumbs-up signals, Scopus logs implicit behavioral signals such as downloads, abstract views, or metadata interactions. These must be interpreted and normalized to serve as effective input for recommendation algorithms.

To address this, we developed an interaction mapping schema that assigns a numerical engagement score, on a scale from 1 to 5, to each type of user action. This mapping was designed to reflect the relative intensity and intent behind each interaction, with higher scores corresponding to more deliberate and content-focused engagement. For example, actions that involve saving or downloading a document were interpreted as strong signals of user interest and assigned the highest score. In contrast, actions such as quick redirection to a publisher's page or metadata browsing were viewed as lower-intent behaviors and assigned correspondingly lower scores.

The mapping schema is structured around five tiers. Score 5 represents high-value actions such as adding a document to a list, saving an alert, or downloading content. These are considered strong indicators of long-term interest or intention to revisit. Score 4 corresponds to meaningful content engagement, including viewing full documents or clicking on titles, which suggests that the user actively consumed or examined the material. Score 3 is associated with moderate engagement such as viewing abstracts, reflecting curiosity or exploratory intent. Score 2 captures lighter interactions, including examination of peripheral metadata like citations or author pages, which indicate interest but not direct content consumption. Score 1 reflects minimal or ambiguous engagement, including actions like viewing content via external links, which may signal a brief or redirected visit rather than focused interest.

This interaction-to-score transformation serves multiple strategic purposes. First, it enables the application of classical recommendation models such as matrix factorization and neural collaborative filtering, which typically rely on explicit feedback signals. Second, it preserves behavioral nuance by differentiating between various levels of user interest, allowing for more fine-grained preference modeling. Lastly, it enhances interpretability and business value by providing a transparent and scalable framework for understanding user engagement across the platform.

Table 3.2: Interaction Mapping Schema

Score	Mapped Interaction Types
5	add to my list, save alert, file downloads, download, content export
4	full document view, title click, content view
3	abstract view, menu-show abstract
2	citations view, author view, related documents
1	view at publisher, full-text via external redirection

3.3. Data Preprocessing

A rigorous data preprocessing pipeline was applied to the raw Adobe Analytics logs to ensure analytical quality and modeling readiness. Given the scale and heterogeneity of behavioral data collected from

the Scopus platform, the preprocessing phase was essential not only for correcting structural inconsistencies but also for maximizing the signal-to-noise ratio in downstream recommendation models.

The process began with the removal of exact duplicate records which typically arise from repeated logging calls or instrumentation noise. Such duplicates can artificially inflate engagement metrics and introduce bias into collaborative filtering models that assume unique user-item interactions. Following this, we addressed a critical integrity issue which was multiple user identifiers assigned to the same session. In total, 216,968 session records spanning January 2024 alone were identified as containing more than one unique user ID per session. These records were removed as preserving them would violate session continuity assumptions and contaminate sequential behavior modeling.

Subsequently, we filtered out all rows containing missing or null values in either the user ID or item ID fields. These identifiers are essential for constructing a reliable interaction matrix, and their absence renders the records unusable for recommendation purposes. Additionally, we excluded records where the `doc_id` field, which is used to identify academic articles, did not match the expected Scopus document ID format. This step ensured that behavioral logs were correctly aligned with the metadata catalog used for content enrichment.

To prevent over-representation of users or items due to repeated activity, we carried out deduplication on user-item interaction pairs. In cases where multiple actions were recorded for the same user and item, we retained only the most valuable interaction based on a predefined engagement hierarchy. For example, a download was considered more meaningful than an abstract view. This approach helped balance how user activity was represented and prevented popular items from dominating the interaction matrix, which could negatively influence collaborative learning outcomes.

As a result of these preprocessing steps, the cleaned dataset for January 2024 was reduced to 40,373,969 rows of high-quality interaction data. This dataset included 95,946 unique users and 2,226,143 distinct academic items. The resulting interaction matrix had a sparsity level of approximately 0.001 percent. These characteristics are in line with real-world academic platforms where users tend to engage with only a small portion of the available content. More than half of the users had fewer than five recorded interactions, highlighting the significance of the cold-start problem and supporting the case for hybrid or content-based recommendation approaches.

When this preprocessing strategy was extended to the entire 2024 calendar year, the final dataset included more than 31.5 million valid interactions involving 674,624 users and 3,775,626 unique items. The average number of interactions per user was 46.76, and the average number per item was 8.35. These values illustrate a pattern of user concentration and long-tail item distribution. The final interaction matrix had a density of only 0.000012 percent, reflecting the extreme sparsity that is typical of academic recommendation environments.

3.4. Knowledge-Enriched Article Graph Construction

To support knowledge-aware recommendation modeling, we extended the core user-item interaction dataset with two additional data layers: (i) a citation graph capturing academic influence among articles, and (ii) a structured item metadata table encapsulating article-level descriptive features. Both were derived from the `ani` dataset, a curated content table representing articles indexed in Scopus. These enriched components were not used for modeling at this stage, but were constructed as part of exploratory data analysis (EDA) to inform and enable future experimentation with graph-based, hybrid, and content-aware recommender systems.

3.4.1. Citation Data Construction

The citation graph was designed to capture the network of academic influence between articles. Using the Scopus-specific `citations` field from the `ani` dataset, we extracted directed citation edges, where each edge represents one article citing another. To ensure academic relevance and data quality, we applied two filters: only articles published from 2014 onwards and those with a minimum of 30 citations were retained. This constraint focused the graph on recent, high-impact publications, aligning with typical business goals such as surfacing authoritative and timely content.

We began with the set of articles actively engaged by users (denoted as set A), then expanded the

citation graph to include one-hop ($A \rightarrow B$) and two-hop ($B \rightarrow C$) citation paths. This approach ensured coverage beyond direct citations, enabling the representation of broader topical or influence-based connections between articles. The resulting graph structure supports downstream models that rely on neighborhood propagation, node connectivity, and academic proximity. After filtering and deduplication, the final citation graph included only article pairs verified to exist in the curated corpus and was stored as `articleRecommendation.article_citation_2024Jan_dense` for experimentation.

3.4.2. Metadata Table Construction

In addition to the citation graph, we constructed a structured metadata table from the `Scopus Metadata` dataset to support content-aware and hybrid recommendation experiments. This table was designed to provide a lightweight yet semantically rich representation of each article, enabling the integration of textual content into knowledge-aware models.

For this purpose, we selectively extracted three key attributes for each article: the **title**, **keywords**, and **abstract**. These fields were chosen because they collectively summarize the core semantic content of an academic paper and are widely used in content-based filtering, embedding generation, and domain-specific representation learning. Unlike more technical metadata such as journal identifiers or author affiliations, the selected fields directly describe the subject matter of the article and offer a generalizable view of its topical relevance.

From a modeling perspective, this metadata enables the use of content-based techniques that rely on textual information, such as generating article embeddings or identifying topic similarities. It allows the recommendation system to go beyond user behavior by considering the actual subject matter of the articles. From a business standpoint, incorporating metadata improves the relevance and transparency of recommendations, making it easier to suggest articles aligned with users' research interests and to explain why specific results were recommended.

3.5. Data Sample Construction for Experimental Evaluation

To conduct reliable and scalable experiments with various recommendation models, we constructed a range of sampled datasets derived from the full interaction data collected in January 2024. These samples were created with two key goals in mind: (i) reducing data size to manageable and representative subsets for modeling, and (ii) controlling for cold-start issues, data sparsity, and domain-specific variability. All datasets were generated from the same source interaction table, and each was designed to serve different evaluation strategies within the knowledge-aware recommendation framework.

3.5.1. Core Sampling Strategy

Core sampling was employed to construct denser and more stable user-item interaction datasets by iteratively removing cold-start users and items. These cold-start entities were defined as those with interactions below a specified threshold. The filtering process was repeated until all remaining users and items met the minimum interaction requirement. This approach produced progressively denser datasets across thresholds ranging from 2 to 6.

Among the sampled versions, Core-2 included over 30,000 users and 112,000 items but continued to exhibit considerable sparsity which limited its effectiveness for robust model training. At the other end of the spectrum, Core-6 yielded a highly dense interaction graph with a density of 0.50 percent but reduced the dataset to fewer than 2,000 users and under 5,000 items which compromised its representational diversity. Intermediate configurations such as Core-3 and Core-4 offered more balanced options yet either remained too sparse or too limited in scale for effective cross-model experimentation.

Core-5 was ultimately selected as the optimal dataset configuration. It contains 2,461 users, 6,099 items, and 43,720 interactions resulting in an interaction density of 0.30 percent. This version provided the best balance between data volume and density. It effectively minimized cold-start noise while preserving a sufficient number of records to support model training, validation, and generalization. The dataset's consistency across both user and item dimensions made it the most suitable foundation for evaluating both baseline and advanced recommendation models.

3.5.2. Domain Sampling

In parallel, we constructed domain-specific samples to enable evaluation of recommendation models within more targeted academic disciplines. Using article metadata, we segmented the interaction dataset into three distinct scientific domains: Computer Science (COMP), Engineering (ENGI), and Environmental Science (ENVI). These domain samples were created to reflect real-world disciplinary contexts where content semantics and user needs differ significantly.

Initially, the extracted domain datasets included large numbers of users and articles, often exceeding feasible limits for iterative experimentation. To address this, we explored multiple filtering strategies aimed at reducing dataset size while maintaining meaningful coverage. These strategies included threshold-based removal of cold-start items, subsampling of users, and density optimization techniques.

After evaluating various configurations, we selected the final domain samples by removing only cold-start items (i.e., those with fewer than 5 interactions), without filtering users. This approach retained a broad user base while ensuring each item was meaningfully represented. The final domain-specific datasets selected for experimentation are summarized in Table 3.3. These versions reflect the optimal balance between dataset size and interaction density, ensuring practical feasibility for model training and evaluation across distinct academic fields.

Table 3.3: Statistics of the Full-Year Dataset (2024)

Dataset	#Interactions	#Users	#Items	Avg/User	Avg/Item	Density (%)
2024_year	58,225,815	696,961	22,208,792	83.54	2.62	0.0004

Table 3.4: Statistics of Partial-Year Datasets (January 2024)

Dataset	#Interactions	#Users	#Items	Avg/User	Avg/Item	Density (%)
core_5	43,720	2,461	6,099	17.77	7.17	0.2913
one_month	2,663,192	95,946	2,226,143	27.76	1.20	0.0012
comp_articles	20,251	7,090	2,670	2.86	7.58	0.1070
envi_articles	15,248	6,110	2,272	2.50	6.71	0.1098
engi_articles	20,903	7,722	3,073	2.71	6.80	0.0881

4

Methodology

4.1. Research Design and Approach

This study adopts a quantitative and experimental research design to evaluate how different recommendation strategies can be applied effectively within a large-scale academic platform. The aim is to systematically assess the utility of collaborative filtering and content-aware models under realistic deployment scenarios, informed by Scopus's diverse user behaviors and technical constraints.

The experimental setup is structured around two core recommendation tasks that reflect the primary use cases of the system. The first is the generation of personalized article recommendations based on a user's past interactions. This mode assumes that the system can access historical engagement data and is optimized to support continuous discovery workflows for returning users. The second task focuses on document-aware recommendation, where related articles are suggested based on a currently viewed target document. This mode does not require user identification and is designed to operate effectively in cold-start or anonymous user sessions.

To implement these tasks, the research is divided into two experimental phases. In the first phase, widely adopted collaborative filtering models such as Bayesian Personalized Ranking (BPR), Factorized Item Similarity Models (FISM), LightGCN, and KGAT are trained using the RecBole framework on an interaction-derived user–item matrix. The objective in this phase is to establish performance benchmarks for personalized top-K recommendations using standard evaluation metrics.

The second phase shifts focus to a hybrid retrieval setting, where recommendations are generated based on either user profiles, document context, or a combination of both. This task better simulates real-world platform behavior, where recommendations must remain relevant whether or not a user is logged in. Evaluation workflows in this phase are designed to isolate model performance in both personalized and non-personalized conditions, providing insights into adaptability and robustness under incomplete information.

Overall, the research approach is tailored to reflect deployment-oriented constraints and aligns with the broader system design goals. It emphasizes methodological rigor through reproducible benchmarking and highlights flexibility by addressing heterogeneous recommendation contexts within a single modular framework.

4.2. Recommendation Generation Strategy

This study proposes and implements a dual-mode recommendation generation framework that includes both *personalized* and *non-personalized* strategies. This design reflects the operational realities of Scopus, where some users have rich interaction histories while others access the platform anonymously or sporadically. By separating these two modes, we ensure that the recommendation system remains effective across a wide spectrum of use cases, such as anonymous browsing, user-specific discovery, and document-level exploration.

The non-personalized strategy focuses on recommending content based solely on the currently viewed article. It does not require any historical user behavior and is thus applicable in scenarios where user identity is not available. In contrast, the personalized strategy utilizes both historical user interactions and the currently viewed article to tailor recommendations to the user's inferred preferences. This blended approach enables the system to address both long-term research interests and session-level contextual intent.

4.2.1. Non-personalized Recommendations

The non-personalized recommendation framework addresses scenarios in which the system lacks access to user-specific interaction history, such as sessions initiated through institutional IPs or external search engines where the user is not logged in. In these cases, the system relies solely on the article currently being viewed to generate relevant recommendations.

After training the recommendation models on user-item interaction data, each article is represented by a fixed-length embedding vector $\mathbf{e}_i \in \mathbb{R}^d$, where d is the embedding dimension. These embeddings encode collaborative, semantic, or citation-based relationships among articles, depending on the architecture used.

All article embeddings are L2-normalized and indexed using the FAISS [13] library to enable efficient similarity search. Given a context article i, the system retrieves the corresponding embedding $\mathbf{e}_i$ and performs top-K retrieval against the entire article corpus. The similarity between articles i and j is computed using cosine similarity:

$$\text{score}(i, j) = \cos(\mathbf{e}_i, \mathbf{e}_j)$$

Since all vectors are L2-normalized, cosine similarity reduces to the inner product, allowing fast approximate nearest-neighbor retrieval using FAISS. The top-K articles with the highest similarity scores are returned as recommendations. Figure 4.1 provides a high-level overview of the end-to-end recommendation generation process in such unauthenticated settings. The process consists of six core stages, each contributing to a scalable and intelligent content discovery mechanism.

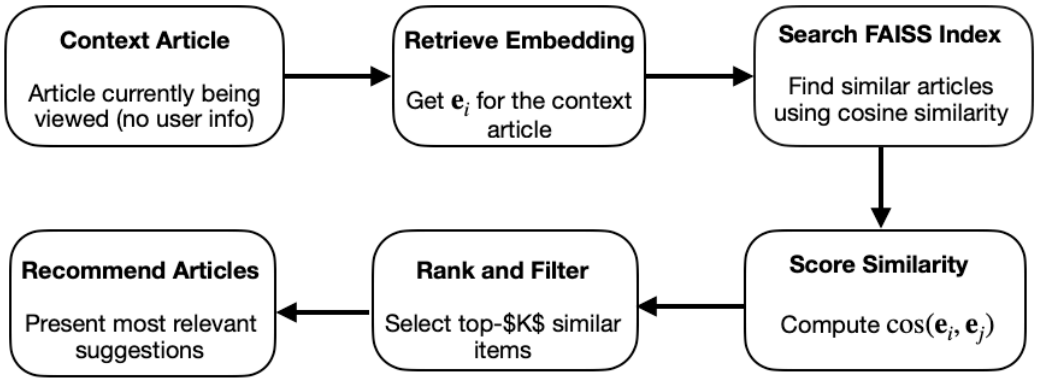


Figure 4.1: Non-personalized recommendation pipeline using item embeddings and FAISS-based similarity retrieval.

This method is particularly effective in the academic domain, where unauthenticated sessions are common. Unlike traditional static methods based on keyword matching, this approach produces thematically aligned and context-aware recommendations grounded in learned representations. It provides a scalable and intelligent fallback mechanism when user signals are unavailable, improving content discoverability and engagement.

An example of non-personalized recommendations generated using this method is presented in Table 4.1.

Table 4.1: Example of non-personalized recommendations retrieved using cosine similarity between item embeddings.

item_id	recommended_item_id	rank	score
92738461529	91527384916	1	0.3219
92738461529	91836475290	2	0.2546
92738461529	91729384615	3	0.2503
92738461529	91283746592	4	0.2330
92738461529	91928374651	5	0.2302

While the retrieval pipeline remains consistent across models, the nature of the embeddings and thus the relevance of the recommendations varies by algorithm. BPR and FISM, although originally designed for personalized ranking, produce item embeddings that capture collaborative co-engagement patterns, making them reasonably effective for similarity-based retrieval. LightGCN refines item embeddings through multi-hop message passing on the user-item graph, enabling the model to incorporate high-order collaborative signals that enhance contextual alignment. KGAT, by contrast, augments item embeddings with structured knowledge from citation graphs via attention-based aggregation. This allows it to retrieve not only co-interacted articles but also semantically or thematically related content grounded in the academic knowledge graph.

4.2.2. Personalized Recommendations

In contrast to the item-only strategy used in non-personalized settings, personalized recommendations leverage historical user interactions to generate tailored suggestions. When a user is logged into the platform, their past interactions with articles such as views, downloads, or citations are used to construct a personalized representation of their long-term research preferences. These interactions are encoded during model training as part of the user-item interaction matrix, resulting in a learned user embedding $\mathbf{e}_u$ that captures collaborative behavioral patterns.

After training the recommendation model (e.g., BPR, FISM, or LightGCN), each user u and each item i are represented by fixed-length embedding vectors $\mathbf{e}_u, \mathbf{e}_i \in \mathbb{R}^d$. These embeddings are optimized to encode latent relationships derived from co-engagement or citation graphs. To incorporate real-time context, the system combines the embedding of the logged-in user with the embedding of the article currently being viewed $\mathbf{e}_c$, forming a hybrid query vector that reflects both long-term preferences and immediate information needs.

The composite query vector is constructed by averaging the two embeddings:

$$\mathbf{q}_{(u,c)} = 0.5 \cdot \mathbf{e}_u + 0.5 \cdot \mathbf{e}_c$$

This query vector is then L2-normalized to ensure consistent scaling and submitted to a FAISS [13] index containing all L2-normalized item embeddings. The similarity between the query vector and each candidate item j is computed using cosine similarity:

$$\text{score}(u, c, j) = \cos(\mathbf{q}_{(u,c)}, \mathbf{e}_j)$$

The top-K items with the highest similarity scores are retrieved and returned as personalized recommendations.

This hybrid approach introduces a flexible and adaptive formulation that fuses static user profiles with dynamic session-based context. Traditional recommendation systems often rely solely on user embeddings, assuming that preferences are stable across time and queries. However, in academic search environments, user interests are often goal-driven and evolve across sessions. By incorporating the embedding of the currently viewed article, this method effectively adjusts to the transient intent of the user while still grounding recommendations in long-term behavioral signals.

To support large-scale evaluation, the recommendation pipeline generates recommendations in batch mode for a set of (u, c) pairs, where each pair consists of a user u and a context article c. For each

pair, the system logs the top-K recommended articles along with their similarity scores and ranks. This output is subsequently used to compute evaluation metrics such as Recall@K, MRR, NDCG, and Hit Rate. Additionally, the output is compatible with LLM-based qualitative assessment pipelines, which assess the semantic relevance and novelty of the recommended articles based on generated summaries or keyword overlap.

Figure 4.2 provides an overview of the personalized recommendation workflow, showing the integration of user and context signals, query formulation, FAISS-based retrieval, and final result generation.

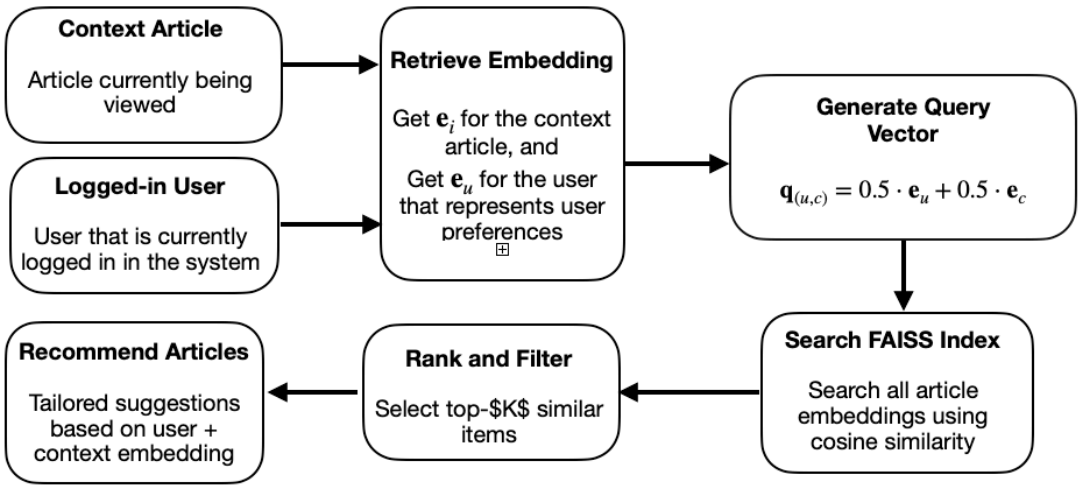


Figure 4.2: Personalized recommendation pipeline combining long-term user preferences with real-time session context.

An example of personalized recommendations generated using this method is presented in 4.2.

4.2.3. Experimental Formulation and Novelty

This recommendation generation setup allows for systematic experimentation across user-based and item-based tasks. Separate retrieval pipelines are implemented for personalized and non-personalized evaluations, allowing models to be tested under both data-rich and data-sparse conditions.

The novelty of this dual-mode strategy lies in its unified use of learned embeddings for both recommendation scenarios. Most academic recommenders focus solely on personalization or rely on keyword-based similarity for document-level suggestions. By implementing a shared embedding space that supports both modes efficiently via FAISS, this work introduces a generalizable and scalable framework suitable for large-scale scholarly platforms like Scopus.

Table 4.2: Example of personalized recommendations retrieved using user interaction history and collaborative filtering.

user_id	viewed_item_id	recommended_item_id	score	rank
73649281547	92738461529	91527384916	0.4123	1
73649281547	92738461529	91836475290	0.3987	2
73649281547	92738461529	91729384615	0.3765	3
73649281547	92738461529	91283746592	0.3552	4
73649281547	92738461529	91928374651	0.3421	5

4.3. Models

This section outlines the recommendation models applied to support both personalized and non-personalized article recommendation tasks. Each model was selected based on its compatibility with implicit feedback, scalability to large datasets, and ability to generalize across different use cases.

The models provide the backbone for computing relevance scores between users and articles or between pairs of articles. After training on user-item interaction data, each model outputs latent embeddings that are used in a unified retrieval pipeline. These embeddings serve as inputs to a similarity-based ranking process, allowing consistent evaluation across both experimental settings.

We focus on three collaborative filtering methods, Bayesian Personalized Ranking (BPR), Factored Item Similarity Model (FISM), and Light Graph Convolutional Network (LightGCN) as well as one knowledge-aware model, the Knowledge Graph Attention Network (KGAT). The following subsections detail each model's theoretical basis and how it was used in the recommendation experiments.

4.3.1. Collaborative Filtering Models

Collaborative filtering is a foundational approach in recommendation systems that relies on the assumption that users with similar interaction patterns in the past will continue to exhibit similar preferences in the future. It leverages historical user-item interaction data to uncover latent structures in user behavior, enabling personalized recommendations without requiring content-level information.

In this study, we explore multiple collaborative filtering techniques to model user engagement with academic articles. Specifically, Bayesian Personalized Ranking (BPR), Factored Item Similarity Models (FISM), and Light Graph Convolutional Networks (LightGCN) were selected as core collaborative filtering approaches. These models were chosen for their compatibility with implicit feedback settings, scalability to large academic datasets, and their complementary strengths in capturing user-item interaction patterns through latent factor modeling and graph-based signal propagation.

Bayesian Personalized Ranking (BPR) Bayesian Personalized Ranking (BPR) is a pairwise ranking optimization algorithm introduced by Rendle et al. [25] to address recommendation problems in implicit feedback scenarios. Unlike traditional pointwise approaches that predict exact preference scores, BPR focuses on the relative ranking of items, aligning more naturally with the implicit nature of interaction data found in platforms like Scopus.

BPR operates on the assumption that users prefer items they have interacted with over those they have not. The learning objective is to maximize the posterior probability that, for any given user u, an observed item i is ranked higher than an unobserved item j. Formally, the optimization objective is:

$$\mathcal{L}_{BPR} = - \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda ||\Theta||^2$$

where D_S denotes the set of all observed user–item–negative item triplets, $\hat{y}_{ui}$ and $\hat{y}_{uj}$ are predicted scores from the model, σ is the sigmoid function, and λ is a regularization coefficient applied to the model parameters Θ to control overfitting. This formulation allows the model to learn latent user and item embeddings through matrix factorization, where each user and item is associated with a vector in a shared latent space. The preference score is computed via the inner product of the respective embeddings.

The BPR model's strengths lie in its simplicity, scalability, and suitability for highly sparse datasets, which are characteristics commonly encountered in academic recommender systems. Given that explicit ratings are rarely available in Scopus, BPR provides an effective mechanism to leverage user behaviors such as downloads, abstract views, or document clicks as implicit signals of interest. Its use of stochastic gradient descent and sampling-based learning allows it to efficiently scale to millions of interactions, making it suitable for large-scale experimentation in real-world academic platforms.

Factored Item Similarity Model (FISM) Factored Item Similarity Model (FISM) is a collaborative filtering algorithm that addresses the item-based recommendation problem by directly learning item-to-item similarities using latent factor representations [15]. Unlike traditional item-based methods that compute similarities through heuristic functions (e.g., cosine or Pearson similarity), FISM learns these relationships through matrix factorization on implicit user feedback.

The core idea behind FISM is to predict a user's preference for a target item by aggregating the latent representations of the items that the user has previously interacted with. More formally, the preference

score $\hat{y}_{ui}$ for user u on item i is estimated as:

$$\hat{y}_{ui} = b_i + \frac{1}{|\mathcal{I}_u|^\alpha} \sum_{j \in \mathcal{I}_u \setminus \{i\}} \mathbf{p}_j^T \mathbf{q}_i$$

where $\mathcal{I}_u$ is the set of items user u has interacted with, $\mathbf{p}_j$ and $\mathbf{q}_i$ are latent vectors for items j and i respectively, b_i is the bias term for item i, and α is a normalization hyperparameter controlling the impact of history length.

One key advantage of FISM is its efficiency in environments with large user-item matrices. Since the model focuses on item-level representations, it avoids maintaining per-user parameters, making it particularly scalable for applications like Scopus, where the number of users can be significantly larger than the number of items. From a business perspective, this translates to faster training and inference time, enabling real-time or near-real-time personalization in large-scale academic platforms.

FISM offers interpretability through item-item relationships, which can be useful for generating recommendations with thematic or semantic coherence. Furthermore, because the model learns from implicit feedback without relying on user metadata, it aligns well with the privacy-preserving data collection practices commonly observed in platforms like Scopus.

In this study, FISM serves as a strong baseline to evaluate how item-level similarities, when learned directly from interaction signals, compare to more complex graph-based or hybrid recommendation strategies. This approach allows us to isolate the effectiveness of similarity-based modeling without additional structural assumptions, providing a clear reference point for assessing improvements introduced by more advanced models.

Light Graph Convolutional Network (LightGCN) Light Graph Convolutional Network (LightGCN) is a state-of-the-art graph-based collaborative filtering algorithm specifically designed for top-N recommendation tasks [11]. It builds upon the success of graph neural networks (GNNs) in recommendation but simplifies the design by removing non-essential components such as feature transformations and nonlinear activation functions. This results in a model that is both computationally efficient and highly effective in leveraging high-order user-item connectivity.

The core idea of LightGCN is to propagate user and item embeddings through the user-item interaction graph. The embedding of a node (user or item) is iteratively updated by aggregating the embeddings of its neighbors. After K layers of propagation, the final embedding $\mathbf{e}_u$ of user u and $\mathbf{e}_i$ of item i are computed by combining the representations across layers:

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \cdot \mathbf{e}_u^{(k)}, \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \cdot \mathbf{e}_i^{(k)}$$

where $\mathbf{e}^{(k)}$ represents the embedding after the k-th propagation layer, and α_k is the layer-wise importance weight (typically set uniformly or learned).

Unlike traditional GNNs that involve transformation matrices and activation functions at each layer, LightGCN solely relies on neighborhood aggregation using normalized adjacency matrices. This not only reduces model complexity but also improves generalization, particularly in sparse datasets such as academic recommendation systems.

In this study, LightGCN is employed to capture multi-hop user-item relationships from interaction data, making it particularly useful for recommendation scenarios with limited direct feedback. From an academic standpoint, it offers an elegant solution to integrating high-order collaborative signals without overfitting. From a business perspective, its scalability and simplified training pipeline make it suitable for deployment in large-scale systems like Scopus, where performance, efficiency, and interpretability are crucial.

Moreover, LightGCN's architecture enables seamless integration with auxiliary information such as citation graphs or metadata by extending the interaction graph to heterogeneous structures, thus providing a foundation for future hybrid and knowledge-aware modeling experiments in this research.

4.3.2. Knowledge Aware Models

Knowledge Graph Attention Network (KGAT) Knowledge Graph Attention Network (KGAT) is a knowledge-aware recommendation algorithm that effectively integrates structured relational information from knowledge graphs into collaborative filtering frameworks using an attention-guided graph neural network architecture [30]. Designed to address the limitations of purely interaction based models, KGAT enables the incorporation of rich semantic information, such as citation relations, domain hierarchies, and entity level metadata, making it especially relevant in academic recommendation scenarios.

At its core, KGAT constructs a heterogeneous graph where both users and items are linked through user item interactions as well as multi relational item item edges derived from a knowledge graph (for example, citation links between academic articles). Each node aggregates information from its neighbors using an attention mechanism that weighs the importance of each relation and neighbor dynamically. The user's preference score for an item is calculated based on the joint propagation of embeddings through both interaction and knowledge structures.

The model is trained using a pairwise learning objective, such as Bayesian Personalized Ranking (BPR) loss, enabling it to distinguish between relevant and irrelevant items based on the learned high-order semantic relationships.

Formally, given an entity e, the attention weight $\alpha_{e,e'}$ between e and its neighbor e' is computed as:

$$\alpha_{e,e'} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{We} \,\|\, \mathbf{We}']\right)\right)}{\sum_{e'' \in \mathcal{N}(e)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{We} \,\|\, \mathbf{We}'']\right)\right)}$$

where $\mathbf{W}$ is a transformation matrix, $\mathbf{a}$ is a trainable attention vector, $\|$ denotes concatenation, and $\mathcal{N}(e)$ is the set of neighboring entities of e.

In the context of this research, KGAT is particularly valuable for leveraging citation data, abstract keywords, and domain-specific context to improve recommendation relevance. KGAT allows us to explore the integration of symbolic knowledge and collaborative patterns, offering a hybrid approach that enriches interpretability and model expressiveness. Such models unlock the potential for semantically enriched personalization, enabling recommendations that reflect topical similarity, scholarly impact, or citation proximity which is capabilities that align closely with the goals of Scopus and similar academic platforms.

KGAT's flexibility in modeling multi-relational graphs also facilitates experimentation with a variety of auxiliary information sources, providing a foundation for future extensions in explainable and domain-aware recommendation systems.

4.4. Evaluation

Evaluation plays a central role in the development and validation of recommender systems, especially in high-stakes environments such as academic publishing, where relevance, trust, and interpretability are of critical importance. Traditional performance metrics such as Recall, NDCG, and Precision have long served as essential tools for assessing ranking performance [7]. Nevertheless, these metrics often fail to reflect more nuanced, user-oriented aspects such as topical novelty, semantic coherence, or contextual appropriateness. These elements are becoming increasingly important for both users and stakeholders in academic discovery platforms [17].

To address these limitations, we adopt a dual evaluation strategy that integrates algorithmic rigor with human-aligned interpretability. First, we implement a traditional offline evaluation protocol using standard top-K ranking metrics via the RecBole framework. This ensures that models are assessed consistently and quantitatively across datasets and algorithms. Second, we introduce a large language

model (LLM)-based evaluation dashboard powered by GPT-4o, which semantically assesses the quality of recommendations in both personalized and non-personalized settings.

This hybrid strategy is particularly suited to Scopus, where user engagement is influenced not only by ranking accuracy but also by how well the recommended articles reflect research themes, contextual novelty, and scholarly intent. As shown in recent literature [24, 18], LLMs offer a powerful semantic validation layer that complements traditional metrics by surfacing subtleties that quantitative evaluation alone may overlook.

4.4.1. Offline Performance Evaluation

Offline evaluation remains the primary method for benchmarking recommender systems in reproducible, scalable environments. By comparing model-generated rankings with held-out user-item interactions, it provides a foundation for quantitative model assessment.

Ground Truth Construction

We adopt RecBole's `leave-one-out` temporal splitting strategy to simulate real-world deployment conditions. For each user, interactions are ordered chronologically: the most recent is used for testing, the second most recent for validation, and the remainder for training.

Personalized Recommendations. In the personalized scenario, each test interaction (u, i_t) represents the ground truth. The recommendation is generated for a user u, potentially contextualized by a previously viewed article i_c, and the system is evaluated on whether the ground-truth item i_t is among the top-K results.

Non-Personalized Recommendations. For anonymous user sessions, a single article i_c is selected as the context item, and other items co-accessed in the same session are treated as implicitly relevant. This item-item co-occurrence provides a reasonable approximation of contextual relevance without user embeddings.

Top-K Ranking Metrics

After ground-truth generation, we evaluate models using standard top-K ranking metrics, all computed at $K = 10$ and averaged across users:

- **Recall@K** quantifies the proportion of relevant items recovered in the top-K list:

$$\text{Recall@}K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\hat{\mathcal{R}}_u@K \cap \mathcal{R}_u|}{|\mathcal{R}_u|}$$

  Recall is crucial in academic discovery, where overlooking relevant literature may negatively affect research completeness.

- **Precision@K** measures the proportion of recommended items that are relevant:

$$\text{Precision@}K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\hat{\mathcal{R}}_u@K \cap \mathcal{R}_u|}{K}$$

  Precision reflects the system's ability to avoid irrelevant or low-quality suggestions.

- **NDCG@K** rewards relevant items that appear higher in the ranked list:

$$\text{DCG@}K = \sum_{i=1}^{K} \frac{rel_{u,i}}{\log_2(i+1)}, \quad \text{NDCG@}K = \frac{\text{DCG@}K}{\text{IDCG@}K}$$

  This metric is particularly appropriate for Scopus, where users typically focus on the top few results.

- **MRR@K** (Mean Reciprocal Rank) captures how early the first relevant item appears:

$$\text{MRR@}K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{rank_u}$$

  Useful in features like "next article to read," where top placements matter most.

- **Hit Rate@K** measures whether any relevant item appears in the top-K:

$$\text{Hit@}K = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}[|\hat{\mathcal{R}}_u @K \cap \mathcal{R}_u| > 0]$$

  This provides a simple yet robust indicator of overall system recall.

Retrieval Workflow

In the personalized setting, we construct a hybrid query vector combining user preferences and session context:

$$\mathbf{q}_{u,i_c} = 0.5 \cdot \mathbf{e}_u + 0.5 \cdot \mathbf{e}_{i_c}$$

The query is L2-normalized and submitted to a FAISS index of all item embeddings. Top-K recommendations are scored using cosine similarity and matched against ground-truth labels.

In the non-personalized setting, the embedding of the context article $\mathbf{e}_{i_c}$ is used as the sole query vector. All recommended items are evaluated against co-accessed articles from the same user session.

All configurations, embeddings, FAISS outputs, and evaluation scores are logged, versioned, and cached to support reproducibility and model comparison.

4.4.2. Language Model-Based Evaluation

Traditional offline metrics such as Recall@K and NDCG serve as essential benchmarks for evaluating recommender systems. However, they do not fully capture the nuanced, human-centered aspects of recommendation quality, especially in academic search settings. In scholarly environments, relevance depends on more than just co-occurrence patterns or click behavior. It involves deeper dimensions such as semantic continuity, interdisciplinary novelty, and the evolving intent behind a researcher's inquiries. To overcome these limitations, we propose a language-model-based evaluation framework that complements classical metrics with rich, interpretable, and semantically grounded assessments generated by large language models (LLMs).

Motivation and System Overview

This evaluation approach is designed to emulate the reasoning of a human expert, such as a research assistant, who can assess how well a recommended academic article aligns with the user's interests and research trajectory. The system operates by prompting Azure OpenAI's GPT-4o model [23] within a scalable Databricks-based infrastructure. Rather than relying solely on indirect behavioral signals like implicit feedback, the framework delivers direct, context-aware evaluations of recommendation quality, expressed through structured natural language explanations.

The LLM-based evaluation is implemented as a modular, model-agnostic pipeline that can operate on outputs from collaborative filtering, content-aware, or hybrid recommenders. It supports three primary evaluation modes:

- **User-based evaluation**, where recommendations are judged in the context of a user's reading history.
- **Document-based evaluation**, where recommendations are compared to a single target article.
- **User profile summarization**, where a user's research interests are synthesized into natural language summaries.

All prompts used in this pipeline, including detailed roles, evaluation criteria, and output formats are included in Appendix A.

User-Based Evaluation

In personalized recommendation scenarios, we simulate a user's academic intent by constructing a synthetic profile using metadata from their prior interactions. This includes keywords and abstract texts from articles they have read. To emphasize central research themes, we apply TF-IDF weighting and remove duplicated terms for clarity. The resulting user profile is then passed into GPT-4o using Prompt 1 (see Appendix A).

In this setting, the LLM evaluates each recommended article along two axes:

- **Relevance (0.0–1.0)** — How closely the article aligns with the user's core research themes and past behavior.
- **Serendipity (0.0–1.0)** — Whether the article provides a surprising but contextually meaningful extension to the user's interests.

The model returns both a JSON-structured score for each dimension and a paragraph-length justification that explains the rationale behind each score. This output is integrated into our analytics dashboards for further inspection, comparison, and iterative model development.

Document-Based Evaluation

For non-personalized scenarios where user history is unavailable or intentionally excluded, we adopt a document-centric evaluation strategy. Here, the goal is to assess whether a recommended article complements or extends the intellectual content of a specific target article.

Using Prompt 2 (Appendix A), the LLM is provided with metadata for both the source and the recommended articles. This includes their titles, abstracts, and keywords. Based on this information, the model is asked to assess how well the recommended article complements or extends the source article in terms of content, methodology, and research focus.

- **Relevance (0.0–1.0)** — The thematic and methodological coherence between the two articles.

As in the user-based setting, the output includes both a numerical score and an evidence-based textual justification.

User Profile Summarization

To improve transparency and interpretability, the system generates natural language summaries that describe each user's academic focus, using Prompt 3 (Appendix A). These summaries are particularly useful for human-in-the-loop evaluations, model diagnostics, and user interface personalization.

The prompt directs the LLM to analyze a curated set of keywords derived from the user's interaction history and to articulate the underlying research themes. Instead of listing topics in isolation, the model organizes related concepts, infers potential research directions, and identifies interdisciplinary connections. The result is a concise and coherent paragraph that reflects the user's academic interests and intellectual scope.

Execution Infrastructure and Optimization

Ensuring scalability and robustness is essential when deploying LLM-based evaluation in environments that require high reliability. To meet this requirement, the system incorporates several architectural enhancements:

- **Asynchronous execution** allows the system to send multiple API requests in parallel, thereby reducing the total processing time for large evaluation batches.
- **Content-aware caching** avoids repeated evaluations by recognizing and reusing identical input cases, which reduces computational load and cost.
- **Error-handling mechanisms** include automatic retries and adaptive request pacing to manage transient failures and external rate limits effectively.

Each response from the language model is parsed into a structured JSON format that contains both numeric evaluation scores and textual explanations. These outputs are stored and visualized through downstream dashboards to support comparison, auditing, and diagnostic workflows.

Strategic Contribution

The evaluation framework presented here marks a conceptual shift from traditional assessment methods. Rather than relying solely on statistical indicators drawn from user behavior, it introduces a cognitively motivated layer of evaluation based on human reasoning and semantic alignment. This approach better reflects how academic users interpret relevance and value in recommendation outcomes.

A defining feature of this framework is its generalizability across various model classes. The same set of prompts can be used to evaluate collaborative filtering algorithms such as BPR and LightGCN,

as well as content-based or hybrid models, without requiring any adjustments. This design simplifies experimentation by enabling consistent evaluation criteria across different systems.

By integrating classical retrieval metrics with semantic scoring from large language models, the framework supports a hybrid evaluation methodology. This dual approach captures both quantitative performance and perceived quality. It enables high-throughput benchmarking while also providing insights that are aligned with user expectations. As a result, the framework closes the gap between system-level optimization and user-centered experience, offering a foundation for future research into explainable and context-aware recommendation evaluation.

Illustrative Output Example

The following examples demonstrate outputs from the LLM-based evaluation pipeline. These results are produced by the system using Prompt 1 (user-based evaluation), Prompt 2 (document-based evaluation), and Prompt 3 (user profile summarization) as documented in Appendix A.

User Profile Summary As part of the user-based evaluation, the system first synthesizes a natural language profile from the user's interaction history. This summary helps contextualize recommendation judgments and enhances explainability:

> *The user specializes in the application of artificial intelligence and machine learning within educational technology and digital transformation. Their work explores the use of AI tools to enhance learning outcomes, student engagement, and assessment design. They are particularly interested in the ethical implications of generative AI in education, with prior experience researching automated feedback systems and adaptive learning platforms. The user has a secondary interest in interdisciplinary innovations that bridge pedagogy with emerging technologies like ChatGPT, and they are exploring frameworks for integrating creative thinking and AI literacy into higher education curricula.*

User-Based Evaluation Output Based on the above user profile, GPT-4o evaluated recommended articles on two axes which are **Relevance** and **Serendipity**. The following table summarizes an example result:

Table 4.3: LLM-generated user-based evaluation output for a recommendation.

item_id	Title	Keywords	Score	Relevance	Serendipity
85181168530	*Future research recommendations for transforming higher education with generative AI*	AI literacy, Assessment, ChatGPT, Generative artificial intelligence, Learning outcomes	0.68	0.62	0.73

Document-Based Evaluation Output In this evaluation mode, a target article is paired with a recommended article, and the LLM provides a **Relevance** score along with an evidence-backed explanation. Table 4.4 below lists two evaluated recommendations, followed by the corresponding semantic explanations generated by GPT-4o.

Table 4.4: LLM-generated document-based evaluation scores.

Target Item ID	Recommended Item ID	Relevance Score
84978121170	85181168530	0.83
84978121170	85180292217	0.72

Explanations:

- **85181168530** — *"The recommended article is highly relevant as it builds on the same thematic foundation of applying AI in educational contexts. Both articles focus on transforming learning environments using AI technologies. While the target discusses adaptive learning with deep learning*

models, the recommended article extends the conversation to generative AI tools like ChatGPT, offering a broader future-oriented perspective. Their shared emphasis on personalization, student outcomes, and digital transformation within higher education strengthens their conceptual alignment."

- **85180292217** — *"The article introduces a related yet distinct perspective by focusing on how generative AI affects student creativity. While not identical in theme, it complements the target article's focus on personalized education through AI, expanding the scope to cognitive and behavioral outcomes. The link between pedagogical personalization and creativity frameworks strengthens the academic continuity."*

4.5. Experimental Framework

This section outlines the experimental framework used to train and evaluate recommendation models across academic datasets. We implemented all experiments using RecBole [37], a widely adopted open-source library designed for standardized recommendation system research. Its modular architecture, built-in support for multiple algorithms, and flexible evaluation pipelines made it a suitable foundation for our experimental workflow.

The goal of this section is to describe the setup used to conduct reproducible and fair comparisons between multiple recommendation models. We begin by detailing our hyperparameter tuning strategy, followed by model-specific training configurations and dataset-aware adjustments. We then present the standardized evaluation settings applied across all models and conclude with an overview of the computational infrastructure leveraged to support training and experimentation at scale.

4.5.1. Hyperparameter Tuning Strategy

To ensure robust and consistent performance across varying data settings and model architectures, we established a comprehensive experimental setup grounded in reproducibility and efficiency. All models were implemented and executed using the RecBole framework, which provided a standardized environment for training, evaluation, and benchmarking. Central to our approach was a systematic hyperparameter tuning strategy, where we performed targeted sweeps over carefully chosen parameter ranges. These ranges extended beyond RecBole's default configurations and included both conservative and aggressive values. This allowed us to capture a broad spectrum of performance behaviors and to identify configurations that optimized learning dynamics and regularization.

Table 4.5: Hyperparameter Search Space for Each Model

Model	Hyperparameters
BPR	`embedding_size` ∈ {16, 32, 64, 128, 256, 1500} `epochs` ∈ {100, 200, 300, 400}
FISM	`embedding_size` ∈ {16, 32, 64, 128, 256, 1500} `epochs` ∈ {100, 200, 300, 400} `alpha` ∈ {0, 0.3, 0.5, 0.7, 1}
LightGCN	`embedding_size` ∈ {64, 128, 256} `n_layers` ∈ {2, 3, 4} `reg_weight` ∈ {1e-6, 1e-5, 1e-4, 1e-3}
KGAT	`embedding_size` = 64 `kg_embedding_size` = 64 `layers` = [64] `mess_dropout` = 0.1 `reg_weight` = 1e-5 `aggregator_type` = 'gcn'

Training Configuration Model training was executed using infrastructure adapted to the computational complexity of each algorithm. For resource-intensive models such as KGAT and LightGCN—particularly on the large-scale one-year dataset—training was conducted using 4 GPUs (`gpu_id: '0,1,2,3'`) with full GPU acceleration (`device: cuda`) to handle graph computations and large-scale embedding

propagation efficiently. In contrast, BPR and FISM, which are less computationally demanding, were trained using CPU resources, which proved sufficient given their matrix factorization-based architectures.

Each model was trained for up to 400 epochs, depending on early stopping behavior, using the Adam optimizer with a learning rate of 0.001. A negative sampling strategy with uniform distribution and a sample size of 1 was employed. The training batch size was set to 8192, and the evaluation batch size to 4096, ensuring scalable processing under memory constraints. Models were evaluated every epoch with a patience threshold of 100 for early stopping.

Evaluation Configuration To ensure consistency across model comparisons, all experiments used the same evaluation configuration:

- `split: {'RS':[0.8,0.1,0.1]}`, grouped by user, ordered by interaction recency.
- `metrics: [Recall, MRR, NDCG, Hit, Precision]` at `topk = 10`
- `valid_metric = MRR@10` with maximization as stopping criterion.

Dataset-Aware Tuning Considerations. Hyperparameter sensitivity varied significantly across datasets with different sparsity levels. For instance, denser subsets such as the COMP domain sample benefited from deeper architectures and higher embedding sizes (e.g., LightGCN with 4 layers and 256 dimensions), while sparser datasets such as the full-year Core 5 sample required stronger regularization and shallower networks to prevent overfitting. These insights were used to adapt model configurations per dataset context during the tuning process.

Infrastructure Description All training experiments were conducted using Amazon Web Services (AWS) SageMaker, with infrastructure tailored to the computational demands of each model. For matrix factorization–based models such as BPR and FISM, which are relatively lightweight in terms of computational complexity, the experiments were executed on `ml.m5.2xlarge` instances featuring 8 vCPUs and 32 GiB of memory. This configuration was sufficient to efficiently train these models on datasets of varying sparsity levels.

In contrast, resource-intensive models such as KGAT and LightGCN, when applied to the full-year 2024 dataset comprising over 31 million interactions, required significantly greater computational power. These models were trained on `ml.g4.12xlarge` instances equipped with 4 NVIDIA A10G GPUs, 48 vCPUs, and 192 GiB of memory. This setup enabled high-throughput parallel training, optimized GPU utilization, and efficient handling of large-scale graph-based computations.

The RecBole framework was extended to support distributed training and batch-level checkpointing, allowing for streamlined experimentation and model reproducibility at scale. This infrastructure design ensured that training and evaluation could proceed efficiently across datasets of varying sizes and sparsity, enabling fair comparison and consistent metric tracking across all models.

5

Experimental Results Analysis

This chapter presents a comprehensive analysis of the experimental outcomes obtained from evaluating multiple recommendation models across a variety of datasets. The goal is to assess both the quantitative ranking performance and the semantic quality of recommendations under different data conditions. The chapter begins by examining personalized and nonpersonalized recommendation results using standard evaluation metrics such as Recall, Precision, NDCG, and Hit Rate. These metrics provide insight into each model's ability to retrieve relevant items within the top K recommendations. The evaluation is then extended using large language model based assessments that capture user relevance, serendipity, and document level semantic alignment, which are aspects often overlooked by traditional metrics. The results are further contextualized by comparing model performance across datasets that differ in sparsity, scale, and domain characteristics. Finally, the analysis highlights the effectiveness of each model in addressing real world challenges such as cold start conditions and scalability, offering insights relevant to practical deployment based on both metric driven and semantic evaluation outcomes.

Each model was trained using the most effective hyperparameter configurations identified through an extensive tuning process. These configurations, which vary according to both the model architecture and the characteristics of each dataset, are presented in **Table 5.3**. The selected settings include factors such as embedding dimensionality, number of graph convolution layers, strength of regularization, and parameters specific to knowledge graph integration. These values were chosen based on validation performance measured through standardized RecBole evaluation procedures, as outlined in the Experimental Setup section. The purpose of this tuning phase was to ensure that each model performed optimally within the context of the specific domain and data characteristics it was applied to.

In this section, we evaluate the effectiveness of two distinct recommendation paradigms: personalized and non-personalized. Personalized recommendations leverage user-specific interaction histories to tailor suggestions, aiming to model individual preferences over time. In contrast, non-personalized recommendations operate solely based on the context of the currently viewed article, making them particularly suitable for cold-start scenarios or anonymous users. Both approaches are assessed under a unified evaluation framework to ensure comparability across datasets and deployment scenarios.

The quality of the personalized and non-personalized recommendations was then evaluated using four standard ranking metrics: **Recall@10**, **Precision@10**, **NDCG@10**, and **Hit Rate@10**. These metrics collectively capture both the presence and the rank position of relevant items within the recommended lists. The results of this evaluation are organized into the following tables:

- **Table 5.1** summarizes Top-10 personalized evaluation results across all models and datasets.
- **Table 5.2** summarizes Top-10 non-personalized evaluation results across all models and datasets.
- **Table 5.4** presents personalized evaluation results of `LightGCN` and `BPR` on the `2024_year` dataset.
- **Table 5.5** presents non-personalized evaluation results of `LightGCN` and `BPR` on the `2024_year` dataset.

- **Table 5.3** details the best hyperparameter configurations per model and dataset, as obtained from tuning.
- **Table 5.6** reports LLM-based evaluation results for *user relevance* across models and datasets.
- **Table 5.7** reports LLM-based evaluation results for *user serendipity* across models and datasets.
- **Table 5.8** reports LLM-based evaluation results for *document relevance* across models and datasets.

To further evaluate the qualitative dimensions of the recommendations, we conducted a semantic-level analysis using large language models (LLMs), focusing on *user relevance*, *user serendipity*, and *document relevance*. These metrics assess how well the recommendations align with user history, introduce novel yet meaningful content, and relate to specific target documents based on metadata such as titles, abstracts, and keywords.

Given the size and richness of the datasets, evaluating all user-item pairs was computationally infeasible. To address this, we applied a sampling-based strategy. For each model and dataset, we randomly selected 50 users to evaluate the semantic similarity between recommended articles and each user's prior interactions. This produced average scores for both user relevance and serendipity. Additionally, we sampled 50 documents per model and dataset, evaluating how closely the recommendations aligned with the content of each document to compute document relevance.

These evaluations were run on AWS Databricks using Apache Spark. We used a `m6gd.4xlarge` driver (64 GB RAM, 16 vCPUs) and up to 10 `m6gd.2xlarge` workers, scaling resources dynamically from 32 to 320 GB of RAM and 8 to 80 vCPUs. This infrastructure supported efficient parallel processing of metadata and recommendation outputs.

The results are reported in **Table 5.6**, **Table 5.7**, and **Table 5.8**, which present average semantic evaluation scores across models and datasets for user relevance, user serendipity, and document relevance, respectively. This LLM-based assessment complements standard metrics by offering a richer understanding of contextual, personalized, and novelty-aware recommendation quality.

Table 5.1: Personalized Recommendation Results (Top-10, in %)

Model	`core_5`	`comp_articles`	`engi_articles`	`envi_articles`
		Recall@10		
FISM	0.85%	2.95%	1.24%	1.90%
BPR	6.93%	7.41%	5.72%	6.00%
LightGCN	7.89%	6.45%	5.90%	5.26%
KGAT	3.37%	1.66%	1.80%	2.53%
		Precision@10		
FISM	0.20%	0.31%	0.14%	0.21%
BPR	1.06%	0.80%	0.62%	0.63%
LightGCN	1.17%	0.86%	0.71%	0.54%
KGAT	4.60%	1.12%	1.28%	1.43%
		NDCG@10		
FISM	0.49%	1.63%	0.54%	0.92%
BPR	5.42%	5.88%	4.15%	4.42%
LightGCN	5.80%	4.12%	3.76%	3.44%
KGAT	5.88%	1.82%	1.97%	2.28%
		Hit@10		
FISM	1.83%	3.10%	1.36%	2.11%
BPR	9.18%	7.86%	6.08%	6.28%
LightGCN	10.28%	7.41%	6.38%	5.36%
KGAT	27.65%	8.51%	9.62%	10.00%

5.1. Performance Comparison Across Datasets

This section presents a comprehensive comparative analysis of recommendation performance across datasets that differ significantly in terms of domain, scale, sparsity, and interaction structure. The em-

Table 5.2: Non-Personalized Recommendation Results (Top-10, in %)

Model	core_5	comp_articles	engi_articles	envi_articles
Precision@10				
FISM	1.02%	0.25%	0.29%	0.09%
BPR	1.02%	0.25%	0.28%	0.09%
LightGCN	1.04%	0.42%	0.41%	0.07%
KGAT	0.29%	0.002%	0.001%	0.14%
Recall@10				
FISM	0.74%	0.54%	0.51%	0.33%
BPR	0.74%	0.54%	0.49%	0.33%
LightGCN	0.74%	1.43%	1.27%	0.66%
KGAT	2.89%	0.003%	0.002%	1.42%
NDCG@10				
FISM	1.35%	0.37%	0.42%	0.23%
BPR	1.35%	0.37%	0.40%	0.23%
LightGCN	1.35%	0.81%	0.76%	0.43%
KGAT	1.36%	0.001%	0.001%	0.62%
Hit@10				
FISM	8.84%	1.90%	2.03%	0.87%
BPR	8.84%	1.90%	2.01%	0.87%
LightGCN	8.84%	4.21%	4.18%	0.66%
KGAT	1.58%	0.002%	0.002%	0.79%

phasis is not solely on contrasting model capabilities in isolation but on understanding how different dataset characteristics influence key performance indicators such as Recall, Precision, NDCG, and Hit Rate. These findings inform practical deployment strategies in both academic and production environments.

Dataset Diversity and Structural Implications The datasets used in this study span a broad range of domains and structural properties. The `core_5` dataset is a relatively dense academic subset, containing a manageable number of users and items with high interaction frequency. In contrast, datasets such as `comp_articles`, `engi_articles`, and `envi_articles` are significantly sparser. These datasets reflect more specialized or expert-level domains where user interactions are limited and item turnover is high.

The `2024_year` dataset presents an industrial-scale scenario with over 22 million items and approximately 58 million user-item interactions. Despite this volume, the dataset remains extremely sparse due to the vast size of the item space. This setup mirrors real-world academic platforms and content recommendation systems, where user engagement is highly uneven, and many items receive minimal feedback. Such datasets are critical for testing the scalability and generalization ability of recommender models under realistic constraints.

Impact of Sparsity and Scale on Recommendation Quality The evaluation results from the personalized setting, shown in Table 5.1 and Table 5.4, clearly illustrate the role of interaction density in driving recommendation performance. The `core_5` dataset consistently produces higher scores across all models. For example, LightGCN achieves a Recall@10 of 7.89%, an NDCG@10 of 5.80%, and a Hit@10 of 10.28%. BPR and FISM also perform well in this setting, though their scores remain slightly lower. This suggests that denser graphs enhance the learning of collaborative signals, resulting in better top-K recommendation quality.

As the datasets become sparser, performance declines sharply. In `comp_articles` and `engi_articles`, all models show a drop in Recall and Precision, reflecting the challenge of learning accurate user preferences when historical data is limited or fragmented. These results confirm that data sparsity acts as a bottleneck to effective personalization. In such settings, recommender systems struggle to distinguish between signal and noise, especially when relying solely on implicit interactions.

Table 5.3: Best Hyperparameter Configuration per Model and Dataset

Model	Dataset	Emb. Size	α	Layers	KG Emb.	Agg.	Dropout	Reg
FISM	core_5	128	0.3	–	–	–	–	–
	comp_articles	32	0.7	–	–	–	–	–
	engi_articles	128	0.0	–	–	–	–	–
	envi_articles	128	0.3	–	–	–	–	–
BPR	core_5	1500	–	–	–	–	–	–
	comp_articles	1500	–	–	–	–	–	–
	engi_articles	1500	–	–	–	–	–	–
	envi_articles	1500	–	–	–	–	–	–
LightGCN	core_5	256	–	3	–	–	–	0.001
	comp_articles	128	–	2	–	–	–	0.0001
	engi_articles	256	–	3	–	–	–	0.0001
	envi_articles	256	–	4	–	–	–	1e-5
KGAT	core_5	64	–	–	64	gcn	0.1	1e-5
	comp_articles	64	–	–	64	gcn	0.1	1e-5
	engi_articles	64	–	–	64	gcn	0.1	1e-5
	envi_articles	64	–	–	64	gcn	0.1	1e-5

Table 5.4: Personalized Evaluation Results of BPR and LightGCN on 2024_year Dataset (Top-10, in %)

Model	Dataset	Precision@10	Recall@10	NDCG@10	Hit@10
LightGCN	2024_year	45.06%	4.57%	24.82%	45.44%
BPR	2024_year	29.74%	3.01%	16.38%	29.82%

The large-scale 2024_year dataset presents a unique case. Personalized evaluation reveals that LightGCN achieves a Precision@10 of 45.06% and a Hit@10 of 45.44%, while BPR attains 29.74% and 29.82% respectively (Table 5.4). These values are substantially higher than those observed in smaller or sparser datasets. The results indicate that even in highly sparse environments, the availability of large volumes of interaction data allows expressive models to generalize well and capture nuanced behavioral patterns.

Personalized versus Non-Personalized Strategies The contrast between personalized and non-personalized settings is substantial. Table 5.2 and Table 5.5 show that performance metrics deteriorate significantly when personalization is removed. For instance, in the non-personalized evaluation on the 2024_year dataset, LightGCN reaches a Recall@10 of only 0.02%, while BPR drops to a similar level. This underscores the critical role of user embeddings and history-based modeling in producing relevant recommendations. Simply ranking items by popularity or item similarity fails to capture the individualized preferences necessary for effective retrieval.

Among the models, LightGCN consistently performs best across both settings, benefiting from its design that leverages higher-order user-item connectivity. In contrast, KGAT exhibits inconsistent behavior. On core_5, it reports a disproportionately high Hit@10 of 27.65%, yet its precision and recall are much lower compared to the other models. This discrepancy suggests a misalignment between ranking and actual relevance, potentially due to noisy propagation of information in the knowledge graph or a lack of fine-grained supervision during training.

Model Performance Scaling and Hyperparameter Sensitivity Performance on the 2024_year dataset highlights the importance of scale. LightGCN capitalizes on the abundance of user interaction data to achieve state-of-the-art performance. The improvement in all top-K metrics confirms that high-volume datasets provide a rich learning signal that enhances model effectiveness. BPR also demonstrates strong performance, validating its continued utility despite its simpler architecture.

The hyperparameter configurations listed in Table 5.3 further illuminate model sensitivity. LightGCN adapts its depth, embedding size, and regularization strength depending on the dataset, which likely contributes to its robust performance. BPR uses a consistently large embedding size of 1500 across all

Table 5.5: Non-Personalized Evaluation Results of `BPR` and `LightGCN` on `2024_year` Dataset (Top-10, in %)

Model	Dataset	Precision@10	Recall@10	NDCG@10	Hit@10
LightGCN	2024_year	0.17%	0.02%	0.18%	1.55%
BPR	2024_year	0.11%	0.02%	0.12%	1.05%

Table 5.6: LLM Evaluation – **User Relevance** (in %)

Model	`core_5`	`engi_articles`	`comp_articles`	`envi_articles`
FISM	38.5%	32.0%	33.5%	7.4%
BPR	42.7%	30.6%	16.4%	13.9%
LightGCN	47.8%	29.1%	8.3%	12.6%
KGAT	13.4%	54.8%	16.0%	9.9%

datasets, indicating that dimensional capacity may compensate for lack of structural complexity. FISM, on the other hand, requires careful tuning of the alpha parameter and embedding size per dataset to remain competitive. KGAT applies the same configuration across datasets, which may explain its inconsistent performance and lack of adaptation to diverse data structures.

Cross-Dataset Summary and Deployment Implications The comparative evaluation clearly indicates that dataset characteristics significantly influence recommendation outcomes. In narrow or sparse domains such as `comp_articles` and `engi_articles`, models struggle to deliver strong performance without personalization or auxiliary features. This highlights the need for hybrid architectures that incorporate semantic embeddings, citation links, or domain knowledge to support collaborative signals. Conversely, in moderately dense settings like `core_5`, even baseline models can achieve acceptable performance, making such datasets useful for benchmarking or transfer learning.

For production-scale deployments similar to the `2024_year` dataset, the evidence supports using graph-based personalized models such as LightGCN. These models not only scale effectively but also maintain high recommendation quality under sparsity constraints. Furthermore, the gap between personalized and non-personalized results reinforces the need for continuous user modeling and dynamic interaction tracking. Finally, the robustness of long-tail retrieval in sparse settings points to the potential of re-ranking layers and diversity-aware interfaces that can surface semantically relevant content from deeper in the recommendation list.

Semantic Evaluation via LLMs: Dataset-Level Trends In addition to classical ranking metrics, we conducted a fine-grained evaluation using large language models (LLMs) to assess recommendation quality along three human-centric dimensions: *user relevance*, *user serendipity*, and *document relevance*. These metrics provide a deeper semantic understanding of how well the recommended items align with users' interests and contextual content cues.

Tables 5.6, 5.7, and 5.8 report these results across all datasets. Notably, the `core_5` dataset yields the most consistent and high-quality LLM evaluation scores, with average user relevance exceeding 38% across models and serendipity reaching over 50%. This can be attributed to its higher interaction density (0.2913%) and narrower domain scope, which enable models to capture clearer user intent signals. Additionally, the stronger user-document alignment (document relevance consistently around 25–28%) reflects the benefit of training on semantically coherent and richly interconnected data.

In contrast, the `engi_articles` dataset demonstrates a different behavior. Despite lower traditional metrics, it exhibits high document relevance across all models (up to 31.5%) and solid serendipity scores (as high as 58.9%). This suggests that, while user profiles may be sparser, the underlying content in this domain is thematically diverse enough to allow recommendation models to generate more surprising yet still relevant items, which is an important trait in knowledge discovery or exploratory academic systems.

The `comp_articles` dataset shows a pronounced drop in both relevance and serendipity scores for several models, highlighting the difficulty of making personalized recommendations in this setting. Its high item sparsity and lower average user interactions (2.86) lead to weaker user modeling and lower

Table 5.7: LLM Evaluation – **User Serendipity** (in %)

Model	core_5	engi_articles	comp_articles	envi_articles
FISM	51.5%	48.1%	48.0%	10.0%
BPR	54.0%	48.6%	23.0%	16.2%
LightGCN	54.6%	44.3%	11.9%	13.4%
KGAT	14.3%	58.9%	22.9%	10.2%

Table 5.8: LLM Evaluation – **Document Relevance** (in %)

Model	core_5	engi_articles	comp_articles	envi_articles
FISM	27.6%	20.7%	27.5%	21.8%
BPR	28.2%	27.5%	23.0%	23.5%
LightGCN	24.9%	30.7%	27.7%	23.1%
KGAT	25.5%	31.5%	22.2%	18.3%

contextual accuracy. Even so, document relevance remains above 22% for all models, suggesting that semantic overlap within the article corpus remains usable, even when user behavior is limited.

Finally, the envi_articles dataset consistently scores the lowest across all LLM-based metrics. Both user relevance and serendipity drop below 14%, and document relevance is generally weaker compared to other sets. This could stem from a combination of sparse feedback, less topical variation, or less discriminative keyword sets in the environmental domain. These findings emphasize the difficulty of training effective recommenders in domains with both semantic ambiguity and interaction sparsity.

Implications for Semantic-Aware System Design. This semantic evaluation complements our traditional metric analysis by highlighting that even when models struggle with top-K precision or recall, they may still deliver contextually meaningful or novel results from a semantic standpoint. Datasets like engi_articles, for instance, appear well-suited for serendipity-aware recommendation systems, while datasets like core_5 benefit from focused user modeling and session-aware algorithms.

These results underscore the importance of aligning dataset structure with business objectives: high-relevance datasets are ideal for precision-sensitive applications (e.g., academic search, news feeds), while serendipitous datasets may be better suited for exploratory systems (e.g., discovery engines, research suggestion tools). Therefore, LLM-based evaluation offers not only an interpretive layer for model diagnostics but also a design signal for tailoring user interfaces, recommendation diversity, and ranking strategies to dataset-specific strengths and limitations.

5.2. Model-Specific Insights

This section provides a comprehensive analysis of model-level performance across diverse datasets. The analysis focuses on four key dimensions: scalability, adaptability to data sparsity, the impact of graph-based signal integration, and broader deployment implications. Each subsection presents detailed observations per model, emphasizing how architectural design and configuration choices affect empirical behavior in practical recommendation settings.

Performance Scaling and Efficiency in Large-Scale Settings

2024_year dataset, containing 58.2 million interactions, around 700,000 users, and 22 million items, is the most computationally intensive environment in this evaluation. It presents significant challenges due to its extremely low interaction density (0.0004%), where traditional collaborative filtering signals are sparse and noisy. Any model deployed in such a setting must be capable of generalizing from minimal interactions while maintaining computational feasibility.

LightGCN demonstrated exceptional performance in this large-scale sparse setting, confirming its scalability and robustness. The model achieved a Precision@10 of 45.06%, Recall@10 of 4.57%, NDCG@10 of 24.82%, and Hit@10 of 45.44%. These results significantly outperform BPR and highlight the effectiveness of LightGCN's simplified architecture, which forgoes non-linearities and uses

only linear neighborhood aggregation in the graph. This design not only reduces computation but also preserves signal strength across multi-hop propagation. Moreover, LightGCN maintained decent performance even in the non-personalized setting, with Precision@10 of 0.17%, indicating its potential for generalized ranking tasks. Its ability to propagate collaborative signals across user-item interaction graphs without overfitting makes it particularly suitable for academic domains where user history is sparse. The LLM-based evaluations further validate this capability: on smaller datasets like `core_5`, LightGCN achieved 47.8% user relevance and 54.6% serendipity, illustrating both semantic fidelity and diversity.

BPR delivered reasonable results on `2024_year`, with a Precision@10 of 29.74%, Recall@10 of 3.01%, NDCG@10 of 16.38%, and Hit@10 of 29.82%. While these results are lower than LightGCN's, they still reflect a strong baseline in personalized recommendation tasks. BPR relies on optimizing pairwise preferences between positive and sampled negative items. This works well when user histories are rich enough to infer preference orders, but in sparse datasets, its reliance on direct user-item pairs limits performance. In the non-personalized setting, BPR drops significantly to 0.11% Precision@10 and 0.02% Recall@10, underscoring its lack of generalization when no user context is available. LLM-based evaluations further revealed its semantic limitations: although BPR maintained moderate relevance and serendipity in some datasets (e.g., `core_5`: 42.7% relevance, 54.0% serendipity), in sparse datasets like `envi_articles` these dropped to 13.9% and 16.2%, respectively. This indicates that BPR tends to favor known or popular items and lacks the capacity for novel or exploratory recommendation.

KGAT was excluded from experiments on `2024_year` due to the overwhelming computational cost of constructing and processing a heterogeneous knowledge graph for 22 million items. To build such a graph incorporating metadata like citations, authors, and keywords would involve billions of typed edges, each needing to be stored, accessed, and dynamically updated during training. This remains beyond the capacity of conventional hardware setups. This limitation is illustrative of a broader challenge in knowledge-aware recommender systems: while metadata-driven methods offer enriched semantic representations, their scaling behavior is constrained unless effective pruning, attention, and sampling strategies are implemented. The lack of evaluation on large datasets for KGAT highlights the urgent need for research on scalable graph construction methods, including edge filtering, dynamic sampling, and hardware-aware representation.

Latent Signal Sensitivity and Dataset Structure Adaptability

The adaptability of models to different structural characteristics of interaction data was tested across four datasets: `core_5`, `comp_articles`, `engi_articles`, and `envi_articles`. These datasets vary in user-item density, average user interactions, and content diversity, offering a diverse benchmark for understanding generalization under sparse, noisy, or semantically narrow conditions.

LightGCN maintained strong generalization across all datasets. In `core_5`, it achieved the highest Recall@10 (7.89%), NDCG@10 (5.80%), and LLM-based relevance (47.8%) and serendipity (54.6%) scores. Its performance held up even in sparse environments such as `engi_articles` (Recall@10 = 5.90%, relevance = 29.1%) and `envi_articles` (Recall@10 = 5.26%). The architecture's reliance on graph-based neighborhood aggregation allows LightGCN to harness even weak signals from distant neighbors. This makes it particularly effective in cold-start settings or domains with limited user engagement. Additionally, LLM-based evaluation revealed that LightGCN recommendations are both semantically relevant and novel, striking a balance between accuracy and diversity. Even in datasets with domain-specific language and limited user overlap (e.g., `comp_articles`), LightGCN outperformed other models in non-personalized and personalized scenarios.

BPR displayed stable performance in mid-density datasets like `comp_articles`, where Recall@10 reached 7.41% and NDCG@10 reached 5.88%. However, in sparser contexts such as `engi_articles` (Recall@10 = 5.72%) and `envi_articles` (Recall@10 = 6.00%), it struggled to maintain semantic quality. In `comp_articles`, BPR achieved moderate user relevance (16.4%) and serendipity (23.0%), but in `envi_articles` these scores fell to 13.9% and 16.2%. This drop indicates that while BPR captures frequent co-occurrence patterns, it lacks mechanisms for semantic generalization. Its bias towards popular or recently interacted items is reflected in its tendency to repeat similar recommendations. Without access to higher-order structural signals, BPR lacks the flexibility to adapt to domains with fragmented user histories or domain-specific vocabulary.

FISM, while strong in co-consumption scenarios, was the most vulnerable to sparsity. In `core_5`, it performed reasonably with 38.5% relevance and 51.5% serendipity. However, in `envi_articles`, its Recall@10 dropped to 1.90% and relevance to just 7.4%. This model relies heavily on direct item-item similarity and assumes that items co-consumed by a user are inherently semantically related. When such overlap is low, the model fails to generalize. It struggles in domains like academic research, where users often access diverse content with minimal historical overlap. As such, FISM is better suited to platforms with repeated and dense consumption behavior (e.g., e-commerce), but not to exploratory or cold-start domains.

KGAT showed erratic behavior across datasets. In `engi_articles`, it surprisingly achieved the highest user relevance (54.8%) and serendipity (58.9%), but in `envi_articles`, both metrics fell sharply to 9.9% and 10.2%. While the model benefits from leveraging heterogeneous metadata (citations, keywords, etc.), its uniform propagation of edge types introduces noise when edge relevance is not properly controlled. The absence of dynamic attention or edge-type weighting mechanisms means that weak or irrelevant connections can distort the learned embeddings. Furthermore, KGAT's reliance on static graphs means it cannot adapt to evolving user interests or domain-specific shifts. The inconsistent performance across datasets suggests that while KGAT can perform well when metadata is clean and dense, it requires architectural refinements to ensure robustness in diverse academic domains.

Graph Integration Effectiveness and KGAT Observations

KGAT's performance introduced a unique contrast within the model cohort. While it underperformed in traditional Top-10 collaborative filtering metrics—particularly on interaction-heavy datasets such as `core_5` and `comp_articles` (Table 5.1 and Table 5.2)—it still demonstrated competitive ranking behavior on smaller and semantically rich datasets like `engi_articles` and `envi_articles`. For instance, in the personalized setting, KGAT achieved Recall@10 of 1.80% on `engi_articles` and 2.53% on `envi_articles`, outperforming FISM and coming close to LightGCN.

However, in non-personalized evaluations, KGAT performed poorly on behaviorally dense datasets, scoring only 0.002% for both Recall@10 and Hit@10 on `engi_articles`, and nearly 0.001% on `comp_articles` (Table 5.2). This sharp drop suggests that without user-specific behavioral signals, KGAT struggles to generate high-quality top-ranked candidates unless guided by strong structural signals.

LLM-based evaluations further emphasized KGAT's potential for semantic understanding. On the domain-specific `engi_articles` dataset, KGAT achieved the highest user relevance (54.8%) and user serendipity (58.9%) among all models, as shown in **Tables 5.6 and 5.7**. It also performed well in document relevance (31.5%, Table 5.8), reinforcing the value of external knowledge propagation in content-aware recommendation contexts.

In contrast, on datasets like `core_5`, which rely heavily on user interactions and exhibit denser behavior logs, KGAT fell behind in both standard and LLM-based metrics. This performance discrepancy indicates that KGAT's graph integration offers the most benefit in domains with structured knowledge, citation networks, or contextual metadata, while offering less advantage when behavioral patterns alone are predictive.

Overall, KGAT emerges as a semantically powerful model, particularly effective in recommendation tasks that benefit from hierarchical relationships and domain-specific content structures. While it may not outperform collaborative filtering baselines in purely behavior-driven contexts, its strength lies in enriching candidate diversity and supporting hybrid pipelines in sparse or metadata-rich environments.

General Recommendations for Deployment and Research Directions

Synthesizing results across both traditional ranking metrics and LLM-based semantic evaluations reveals important directions for recommender system deployment and future research. Among all models evaluated, LightGCN consistently emerges as the most effective and adaptable framework for academic recommendation tasks. It combines high ranking accuracy with strong semantic alignment, maintaining top performance across both sparse and dense datasets. Its simple yet powerful architecture avoids non-linear components in favor of linear graph aggregation, resulting in a model that is both computationally efficient and robust at scale. These characteristics make it well-suited for industrial deployment, particularly in environments where data sparsity and long-tail item distributions are

prevalent.

In contrast, BPR continues to serve as a reliable baseline, especially in mid-density settings where pairwise ranking is sufficient to capture user preferences. However, its performance tends to plateau in sparse scenarios due to its limited capacity to model higher-order interactions. Additionally, it often emphasizes historically popular items, which can lead to semantically narrow and repetitive recommendations. To remain competitive in more complex domains, BPR requires the integration of diversification strategies, including hybrid scoring methods, reranking modules, or contextual information.

FISM demonstrates its greatest effectiveness in datasets that exhibit strong co-consumption behavior. While it delivers acceptable performance in dense interaction scenarios, its item-similarity design does not generalize well to environments with sparse or irregular user activity. As evidenced by the LLM-based evaluations, FISM frequently underperforms in terms of user relevance and serendipity. This limits its utility to domains where user behavior is stable and repetitive over time, which is rarely the case in academic or exploratory recommendation settings.

KGAT presents a conceptually appealing approach due to its use of knowledge graph metadata. However, its practical utility remains limited. The model's uniform edge propagation and static attention mechanisms often dilute important semantic signals, leading to inconsistent performance across datasets. Although KGAT demonstrated strong results in specific scenarios, such as the `engi_articles` dataset, these successes were not reliably replicable. Its substantial computational demands also hinder scalability to larger datasets, such as `2024_year`, pointing to the need for further refinements. Future improvements could include edge-aware attention mechanisms, noise filtering techniques, and more efficient graph sampling strategies.

In conclusion, the future of recommender systems depends on balancing predictive accuracy with semantic richness and user-centered relevance. LightGCN provides a strong foundation for building scalable and semantically meaningful recommendation pipelines. Continued research into hybrid models and graph-based enhancements will be essential for advancing personalization and discovery in academic and domain-specific contexts.

5.3. Comparative Evaluation of Keyword-Based and LightGCN-Based Recommendation Systems

To extend the evaluation beyond personalized recommendation pipelines, we conducted a focused analysis comparing the performance of the current production recommender system in Scopus with a non-personalized adaptation of LightGCN. This comparison was motivated by the practical constraint that Scopus currently does not offer personalized recommendations tied to user profiles. As such, the evaluation investigates the relative effectiveness of two item-to-item recommendation strategies: one based on keyword overlap and the other grounded in collaborative filtering, both assessed against historical user interaction data.

The existing Scopus recommender relies on content-based filtering, where related articles are retrieved using similarity scores computed over document metadata such as titles, abstracts, and controlled terms. This approach leverages domain-specific lexical patterns, but it does not incorporate any user behavior signals. As a result, it lacks the capacity to reflect nuanced user preferences or uncover latent relationships between documents that go beyond surface-level textual similarity.

By contrast, LightGCN is a collaborative filtering model that learns item representations via multi-hop propagation on a user-item bipartite graph. Although commonly used for personalized recommendations, LightGCN can be adapted for non-personalized use by generating document-to-document similarity scores based on item embeddings alone. This enables behavior-informed, identity-agnostic recommendations, even when user-specific data is unavailable. Through this comparison, we assess whether the collaborative signals captured by LightGCN result in more behaviorally relevant and semantically novel suggestions than those produced by the existing keyword-based system. The outcomes of this evaluation provide valuable insights into the feasibility and potential impact of integrating LightGCN into Scopus's recommendation architecture, especially as a stepping stone toward future hybrid or personalized systems.

Evaluation Design and Input Preparation

To compare the keyword-based and LightGCN-based non-personalized recommendation systems, we prepared two evaluation pipelines operating over a common set of target documents. Each system returned a Top-5 ranked list of recommendations per target item.

The keyword-based system, derived from the current Scopus production recommender, provided five fixed recommendations based on lexical similarity. These were initially stored in wide format across multiple columns (`rec_1` to `rec_5`) and were transformed into long format using PySpark's `stack` function to facilitate row-wise comparisons.

The LightGCN recommendations were stored as a JSON array containing recommendation-document pairs with associated scores. We parsed these arrays using a structured schema and exploded the arrays to extract the top five documents per target item.

To determine relevance, we used implicit feedback derived from a historical user-item interaction table. Specifically, a recommended document was considered relevant if it had been previously interacted with by a user who also interacted with the corresponding target document. This co-engagement was treated as a binary relevance signal. We then computed evaluation metrics by joining recommendation outputs with this interaction-derived ground truth.

Each recommendation list was assigned ranks from 1 to 5. To calculate NDCG, a logarithmic discounting function was applied to each ranked recommendation, with hits contributing $\frac{1}{\log_2(\text{rank}+1)}$. Precision, Recall, HitRate, and NDCG were computed per target document and averaged across the test set to yield global performance scores.

Metrics and Evaluation Procedure

After assigning binary hit labels to each recommendation, we computed four standard ranking metrics.

- **Precision@5:** Proportion of the five recommended items that were labeled as relevant based on user co-engagement.
- **Recall@5:** Identical to Precision@5 in this setup, due to the use of a fixed number of relevance opportunities per target document.
- **HitRate@5:** Binary indicator denoting whether at least one relevant item was retrieved for a given target document.
- **NDCG@5 (Normalized Discounted Cumulative Gain):** Measures not just the presence of relevant documents, but also their positions in the ranked list, with higher weights assigned to items that appear earlier:

$$\text{NDCG@5} = \sum_{i=1}^{5} \frac{\text{hit}_i}{\log_2(i+1)}$$

These metrics were computed at the target-document level and subsequently averaged over the entire test set to produce aggregate scores for each system.

Results and Comparative Performance

Table 5.9 presents the aggregated evaluation results for both recommendation methods. All metrics reflect averages over the full test set of target documents.

Table 5.9: Evaluation Metrics for Non-Personalized Recommendations

Metric	Keyword-Based System	LightGCN-Based System
Precision@5	2.60%	36.47%
Recall@5	2.60%	36.47%
HitRate@5	6.84%	71.50%
NDCG@5	6.52%	99.08%

The LightGCN-based recommendation system outperforms the keyword-based system by a substantial margin across all metrics. Precision@5 and Recall@5 improved from 2.60% to 36.47%, showing that

LightGCN is significantly better at retrieving behaviorally relevant documents. HitRate@5 increased more than tenfold, confirming that LightGCN retrieved at least one relevant document in over 70% of the target cases. NDCG@5 rose from 6.52% to 99.08%, indicating that relevant documents recommended by LightGCN were ranked consistently near the top.

These results emphasize the limitations of relying solely on keyword or metadata similarity. The keyword-based approach fails to model actual user behavior, leading to low relevance and engagement. In contrast, LightGCN leverages collaborative signals from user-item interactions to recommend articles that are not only semantically aligned but also behaviorally grounded.

The dramatic performance gains observed in this evaluation suggest that collaborative filtering methods such as LightGCN are highly effective for non-personalized recommendation scenarios. Even without access to user-specific context, LightGCN captures meaningful item-to-item relationships that reflect real-world engagement patterns.trajectories.

Implications for Scopus Recommendation Architecture

This analysis illustrates the significant limitations of content-only recommenders in capturing the dynamic and behavior-driven nature of academic discovery. Although the current keyword-based system surfaces documents with high textual similarity, it fails to align with actual user behavior, resulting in a low retrieval rate of relevant content.

By leveraging historical user-item interactions, the LightGCN-based recommender is able to surface documents that are not only topically aligned, but also behaviorally grounded. Even in a non-personalized setting, this method captures latent preferences and higher-order relationships that are not accessible through keyword overlap alone.

These findings suggest a clear direction for enhancing Scopus's recommendation infrastructure. Integrating collaborative filtering models such as LightGCN would allow for richer and more adaptive recommendations that reflect real-world usage patterns. While this experiment was conducted in a non-personalized setting, the performance gains indicate that collaborative embeddings can serve as a strong foundation for more advanced systems that incorporate both user and content signals.

6

Limitations

This study provides valuable insights into the design and evaluation of personalized academic recommendation systems using interaction data from Scopus. However, several limitations emerged throughout the experimentation process. These limitations are primarily rooted in the nature of the dataset and the constraints of the computational environment. Acknowledging these limitations is essential for contextualizing the results and identifying opportunities for future improvement.

6.1. Data-Centric Limitations

A central limitation of this study arises from the characteristics of the user behavior observed on the Scopus platform. Unlike general-purpose platforms such as Netflix or Amazon, where users engage frequently and leave explicit feedback, Scopus users typically access the platform with specific and immediate research objectives. Their interactions are often limited to viewing abstracts, downloading full texts, or checking references, which are short-term and highly focused behaviors.

This results in a very sparse user-item interaction matrix, where most users engage with only a few articles. The limited volume of interactions per user reduces the amount of collaborative information available for learning user preferences. Additionally, Scopus does not support explicit ratings such as likes or scores. As a result, this study relies on implicit interaction signals, such as article views, to infer user interest. These signals can be noisy and do not always reflect a user's true level of satisfaction or engagement.

The reliance on implicit feedback and sparse user histories limits the effectiveness of models that depend on rich user preference signals. In comparison to datasets such as MovieLens [8], Amazon Reviews [9], or Yelp [34], which provide more abundant and structured user feedback, the Scopus dataset imposes significant challenges in capturing nuanced user behavior. Consequently, model performance on Scopus tends to be lower than reported in other research domains that benefit from more expressive training signals.

6.2. Infrastructure and Computational Constraints

Another key limitation of this work relates to infrastructure and resource availability during experimentation. All experiments were conducted using a cloud-based environment hosted on Amazon Web Services. For training large-scale models such as KGAT and LightGCN on the full one-year dataset, we used a high-performance instance with multiple GPUs. For less demanding models such as BPR and FISM, a CPU-based instance was used.

The use of a single-node setup introduced a number of constraints. Because the entire GPU or CPU capacity was occupied during the training of large models, it was not possible to run multiple experiments at the same time. This significantly increased the total time required to complete all training and evaluation tasks.

Furthermore, debugging memory usage and optimizing data pipelines for large datasets were time-

consuming tasks that required careful tuning. The high cost of scaling up infrastructure prevented the use of parallelized hyperparameter search or model ensemble techniques. The trade-off between cost efficiency and computational performance had to be managed carefully throughout the project timeline.

Knowledge Graph Limitations and Metadata Underutilization

A significant limitation in this study pertains to the underutilization of available article metadata and the restricted relational scope of the constructed knowledge graph. Although Scopus provides rich metadata such as abstracts, titles, keywords, journals, and authorship, these features were not fully exploited in the collaborative filtering models. Particularly in sparse academic environments, where user-item interactions are limited, integrating textual semantics and contextual signals is critical to overcoming cold-start problems and improving recommendation accuracy.

Initial efforts were made to enrich the knowledge graph by incorporating metadata-based relations alongside user-item interactions and citation links. For instance, we explored graph extensions where articles were linked based on semantic embeddings derived from abstracts and keywords. However, this approach significantly increased the graph's dimensionality and density. The resulting computational overhead made training prohibitively slow even for a single epoch on large-scale datasets, given the limited availability of high-performance GPU nodes.

Moreover, the citation-based graph used in the final KGAT implementation lacked other important relational types such as topical similarity, co-authorship, and shared journal domains. This restricted KGAT's ability to capture higher-order dependencies and nuanced semantic relationships among articles. As a result, while citation edges offer a foundational signal of academic relevance, they fall short of modeling the multifaceted structure of scholarly knowledge.

These challenges were compounded by time and budget constraints that precluded further exploration of multi-relational or hierarchical graph structures. Nevertheless, our observations highlight the value of high-quality, semantically enriched knowledge graphs for academic recommender systems. Future implementations should consider scalable approaches to integrate metadata more effectively, possibly through pre-computed content embeddings, graph pruning strategies, or hybrid recommendation architectures that combine collaborative and content-based signals.

7

Conclusion

This thesis presented the design, implementation, and evaluation of a scalable, modular, and intelligent academic recommendation framework tailored for Scopus. It addressed a core limitation in existing academic discovery tools: the inability to adapt recommendations based on user behavior or content semantics. By developing a dual-mode retrieval pipeline that supports both personalized and non-personalized strategies, the system bridges the gap between engagement-driven and content-driven recommendation approaches.

The system architecture featured a two-stage pipeline comprising candidate generation and reranking components. Four collaborative filtering models (FISM, BPR, LightGCN, KGAT) were evaluated for candidate generation across multiple datasets, each characterized by varying levels of sparsity, scale, and topical focus. Reranking incorporated citation-based features and journal quality indicators to align recommendations with scholarly relevance signals.

Key experimental findings highlighted the consistent superiority of personalized recommendation methods, particularly LightGCN, across datasets with sufficient user interaction history. LightGCN achieved strong performance in both ranking quality and computational efficiency, demonstrating robustness even in datasets with hundreds of thousands of users and millions of items. Its effectiveness reflects the capacity of graph-based collaborative filtering to capture user-item relational patterns in sparse feedback environments.

On the other hand, non-personalized strategies played a critical role in cold-start scenarios where user histories were absent or minimal. These include first-time users, anonymous sessions, or domains with insufficient historical activity. In such cases, citation-based similarity and keyword matching offered reasonable alternatives, though with lower personalization depth. The coexistence of both approaches within a unified pipeline allows the system to support a wide range of user contexts, which is essential for the diverse Scopus user base.

An important methodological contribution of this thesis is the incorporation of large language model-based semantic evaluation. Standard top-K ranking metrics such as Recall, NDCG, Precision, and Hit Rate were extended with qualitative assessments of relevance and serendipity generated through GPT-4-based evaluation. This added a semantic dimension to the evaluation process, allowing for the identification of thematic alignment and conceptual novelty that may not be reflected in ranking accuracy alone. For instance, KGAT exhibited strong performance in thematic continuity, despite yielding only moderate scores in traditional metrics.

In addition, this research emphasized the value of exploratory data analysis (EDA) in system design. Dataset profiling based on interaction density, user activity patterns, and citation structure informed the selection of modeling strategies, sparsity handling techniques, and evaluation design. EDA also uncovered behavioral signals, such as disparities in download and save frequencies or the presence of inactive user clusters, which can inform future efforts in user segmentation and adaptive modeling.

The framework was benchmarked against Scopus's existing keyword-based recommendation engine

in a real-world scenario. LightGCN significantly outperformed the baseline across all ranking metrics and has been selected for production deployment to generate recommendations for registered users with sufficient historical interactions. This integration represents a transition toward behaviorally informed recommendation logic. For new or anonymous users, the system retains the ability to generate non-personalized recommendations using scalable similarity-based methods, which are currently undergoing further refinement.

Several directions remain for future development:

- Integration of article-level metadata such as abstracts, keywords, and publication details into the model architecture could improve performance in sparse datasets and support cross-domain generalization.
- Validation and weighting of user interaction signals, including views, saves, and downloads, would enhance the system's understanding of engagement and could be achieved through ablation studies or online A/B testing.
- Enriching user profiles with external academic data, such as publication records, institutional affiliations, and co-authorship networks, could enable personalized recommendations that adapt to the user's academic career stage and research focus.

These future developments are essential for further enhancing the system's ability to deliver accurate, contextually aware, and user-tailored recommendations. Integrating rich article-level metadata would allow the models to move beyond collaborative filtering's reliance on interaction histories and improve performance in sparse data scenarios while enabling cross-disciplinary knowledge transfer. Validating and optimally weighting user interaction signals ensures that the system learns from the most informative engagement patterns and improves the reliability of its predictions while reducing noise from low-intent behaviors. Enriching user profiles with external academic information would provide deeper personalization by allowing the system to distinguish between user types, research domains, and career stages, ultimately supporting a more targeted, relevant, and impactful discovery experience for every researcher.

In summary, this thesis contributes a flexible and production-ready academic recommendation framework that balances scalability, personalization, and semantic relevance. It combines collaborative filtering with domain-aware recommendations and introduces large language model-based evaluation as a complementary metric for assessing recommendation quality. Validated on real-world Scopus datasets and designed for deployment, the system advances the current state of academic discovery by aligning technical performance with user-centric goals. As academic information systems continue to grow in scale and complexity, such adaptive and intelligent recommendation tools will be critical in enhancing research accessibility and relevance.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (2005), pp. 734–749.

[2] Xiaomei Bai et al. “Scientific paper recommendation: A survey”. In: *Ieee Access* 7 (2019), pp. 9324–9339.

[3] Joeran Beel and Bela Gipp. “BERTopicRec: Enhancing Academic Paper Recommendations Using BERT-based Topic Modeling”. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*. 2022.

[4] Joeran Beel et al. “Paper recommender systems: a literature survey”. In: *International Journal on Digital Libraries* 17.4 (2016), pp. 305–338.

[5] Arman Cohan et al. “SPECTER: Document-level Representation Learning using Citation-informed Transformers”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 2270–2282.

[6] Andrew Collins and Joeran Beel. “Document embeddings vs. keyphrases vs. terms for recommender systems: a large-scale online evaluation”. In: *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE. 2019, pp. 130–133.

[7] Asela Gunawardana and Guy Shani. “A survey of accuracy evaluation metrics of recommendation tasks”. In: *Journal of Machine Learning Research* 10.Dec (2009), pp. 2935–2962.

[8] F. Maxwell Harper and Joseph A. Konstan. “The MovieLens Datasets: History and Context”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.4 (2015), pp. 1–19. DOI: 10.1145/2827872.

[9] Ruining He and Julian 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. DOI: 10.1145/2872427.2883037.

[10] Ruining He and Julian McAuley. “VBPR: visual bayesian personalized ranking from implicit feedback”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[11] Xiangnan He et al. “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 (SIGIR)*. ACM, 2020, pp. 639–648. DOI: 10.1145/3397271.3401063.

[12] Weizhi Ji et al. “A Survey on LLMs for Recommendation: Taxonomy, Methods, Applications and Open Challenges”. In: *arXiv preprint arXiv:2307.10174* (2023). URL: https://arxiv.org/abs/2307.10174.

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. *Billion-scale similarity search with GPUs*. 2019. arXiv: 1702.08734 [cs.CV]. URL: https://arxiv.org/abs/1702.08734.

[14] Santosh Kabbur, Xia Ning, and George Karypis. “Fism: factored item similarity models for top-n recommender systems”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 659–667.

[15] Srikumar Kabbur, Xia Ning, and George Karypis. “FISM: Factored Item Similarity Models for Top-N Recommender Systems”. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2013, pp. 659–667. DOI: 10.1145/2487575.2487591.

[16] Yehuda Koren. “Factorization meets the neighborhood: a multifaceted collaborative filtering model”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 426–434.

[17] Karl-Heinz Krempels and Martin Rajman. “Beyond accuracy: Evaluating recommender systems by their impact”. In: *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. 2022, pp. 155–163.

[18] Xiaotian Liu et al. “LLM4Rec: Empowering Large Language Models for Enhanced Recommendation Quality Assessment”. In: *arXiv preprint arXiv:2310.06878* (2023).

[19] Yuxuan Liu et al. “User-aware Graph Collaborative Filtering for Cold-Start Academic Paper Recommendation”. In: *Proceedings of the 44th International ACM SIGIR Conference*. 2021.

[20] Félix Moya-Anegón et al. “Scopus: A bibliometric database”. In: *Proceedings of the 17th International Conference on Scientometrics and Informetrics* (2019), pp. 1025–1031.

[21] Xia Ning and George Karypis. “Slim: Sparse linear methods for top-n recommender systems”. In: *2011 IEEE 11th international conference on data mining*. IEEE. 2011, pp. 497–506.

[22] Shumpei Okura et al. “Embedding-based news recommendation for millions of users”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 1933–1942.

[23] OpenAI. *GPT-4o Technical Report*. Accessed August 2025. 2024. URL: `https://openai.com/index/gpt-4o`.

[24] Ashwin Paranjape, Mohammad Kachuee, and Zachary C Lipton. “Improving semantic coherence in recommender systems using Sentence Mover’s Similarity”. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 791–799.

[25] Steffen Rendle et al. “BPR: Bayesian personalized ranking from implicit feedback”. In: *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press. 2009, pp. 452–461.

[26] Steffen Rendle et al. “BPR: Bayesian personalized ranking from implicit feedback”. In: *arXiv preprint arXiv:1205.2618* (2012).

[27] Francesco Ricci, Lior Rokach, and Bracha Shapira. “Introduction to recommender systems handbook”. In: *Recommender Systems Handbook*. Springer, 2011, pp. 1–35.

[28] Haoran Sun, Ying Liu, and Xiangnan He. “ChatGPT as a Recommender System Evaluator: A Case Study in Academia”. In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2023, pp. 3206–3210. DOI: `10.1145/3539618.3591720`.

[29] Wenhao Sun et al. “Scaling Graph-Based Recommendation with Simplified Graph Convolutional Networks”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[30] Xiang Wang et al. “KGAT: Knowledge Graph Attention Network for Recommendation”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. ACM, 2019, pp. 950–958. DOI: `10.1145/3292500.3330989`.

[31] Xiang Wang et al. “Neural graph collaborative filtering”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 165–174.

[32] Yuxuan Wang, Wei Liu, and Yiming Chen. “DeepCF-HT: Deep Collaborative Filtering with Heterogeneous Textual and Categorical Features”. In: *Information Sciences* 626 (2023), pp. 277–294.

[33] Jie Wu et al. “GCRec: Graph Collaborative Filtering for Scholarly Paper Recommendation”. In: *Journal of Information Science* 48.3 (2022), pp. 372–389.

[34] Shuai Zhang et al. “A Survey on Deep Learning-Based Recommender Systems: From Collaborative Filtering to Content Augmentation”. In: *ACM Computing Surveys (CSUR)* 54.1 (2021), pp. 1–38. DOI: `10.1145/3439727`.

[35] Shuai Zhang et al. “Deep learning based recommender system: A survey and new perspectives”. In: *ACM Computing Surveys (CSUR)* 52.1 (2019), pp. 1–38.

[36] Wei Zhang, Han Chen, and Zhipeng Zhao. “Content-aware neural collaborative filtering for scholarly recommendation”. In: *Proceedings of the 44th International ACM SIGIR Conference*. 2021.

[37] Wayne Xin Zhao et al. “Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms”. In: *proceedings of the 30th acm international conference on information & knowledge management*. 2021, pp. 4653–4664.

A

LLM-Based Evaluation Prompts

This appendix provides the full prompt templates used for evaluating academic article recommendations using GPT-4o via Azure OpenAI. Each prompt has been tailored for a specific evaluation mode: user-based, document-based, and profile summarization.

Prompt 1: User-based Evaluation

```
<ROLE>
You are a research assistant helping evaluate academic recommendations for users based on their reading history and research interests.
</ROLE>

<TASK>
Evaluate how well a recommended article fits a 'users intellectual interests and reading behavior.
</TASK>

<USER PROFILE>
- Full Keywords: {', '.join(user_profile.get('all_keywords', []))}
- Total Interactions: {user_profile.get('interaction_count', 0)}
</USER PROFILE>

<RECOMMENDED ARTICLE>
- Title: {rec_doc.get('title', 'N/A')}
- Abstract: {rec_doc.get('abstract', 'N/A')[:300] if isinstance(rec_doc.get('abstract', 'N/A'), str) else 'N/A'}
- Keywords: {rec_doc.get('keywords', 'N/A')}
</RECOMMENDED ARTICLE>

<EVALUATION CRITERIA>
1. Relevance (0.0 - 1.0) - ...
2. Serendipity (0.0 - 1.0) - ...
</EVALUATION CRITERIA>

<INSTRUCTIONS>
Return your evaluation in JSON format.
...
</INSTRUCTIONS>

<OUTPUT FORMAT>
{
  "relevance": <float>,
  "serendipity": <float>
}

Relevance: ...
Serendipity: ...
```

Prompt 2: Document-based Evaluation

```
<ROLE>
You are a research assistant evaluating the relationship between two academic articles...
</ROLE>

<TASK>
Assess how well a recommended academic article complements or extends the target article...
</TASK>

<TARGET ARTICLE>
...
</TARGET ARTICLE>

<RECOMMENDED ARTICLE>
...
</RECOMMENDED ARTICLE>

<EVALUATION CRITERIA>
1. Relevance (0.0 - 1.0) - ...
</EVALUATION CRITERIA>

<INSTRUCTIONS>
Return your evaluation in JSON format...
</INSTRUCTIONS>

<OUTPUT FORMAT>
{
  "relevance": <float>
}

Relevance: ...
```

Prompt 3: User Profile Summary

```
<ROLE>
You are an expert academic advisor with a deep understanding of interdisciplinary research
    areas.
</ROLE>

<TASK>
Based on a set of keywords extracted from academic articles a user has read...
</TASK>

<CONTEXT>
These keywords were gathered from article metadata...
</CONTEXT>

<KEYWORDS>
{', '.join(sorted(keywords))}
</KEYWORDS>

<INSTRUCTIONS>
Group related concepts, avoid listing, synthesize...
</INSTRUCTIONS>

<OUTPUT FORMAT>
A single paragraph describing the 'users research interests.
```