AI Based Course Suggestion Portfolio

Ritik Raushan
Department of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, Gautam Buddh Nagar, India
ritik172002@gmail.com
Sangam Kumari
Department of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, Gautam Buddh Nagar, India
kumarisangam9262@gmail.com
Sanskriti Singh
Department of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, Gautam Buddh Nagar, India
Sanskritisingh1922@gmail.com
Pawan Mishra
Department of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, Gautam Buddh Nagar, India
Pawan.it@gniot.net.in
Prof.Dr.Vikas Singhal
Professor & Head, Department of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, Gautam Buddh Nagar, India
profvikassinghal@gmail.com
Dr.Ajay Sahu
Professor, Department of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, Gautam Buddh Nagar, India
ajaysahu.it@gniot.net.in

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.11/Issue12/DCAE10082
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.11/Issue12/DCAE10082
Received:10,November 2024,Revised: 21,November 2024,Accepted: 30, November 2024, Published Online: 13, December 2024. https://www.ijirae.com/volumes/Vol11/iss-12/03.DCAE10082.pdf
Article Citation: Ritik, Sangam, Sanskriti, Pawan, Prof.Dr.Vikas, Dr.Ajay (2024), AI Based Course Suggestion Portfolio. IJIRAE:: International Journal of Innovative Research in Advanced Engineering, Volume 11, Issue 11 of 2024 pages 847-853 **Doi:>** https://doi.org/10.26562/ijirae.2024.v1112.03
BibTeX Key: Ritik@2024AI

Abstract: The rapid expansion of online education platforms has fundamentally changed the way students engage with learning opportunities. Students now have access to a vast array of courses, offered by different institutions, in various fields of study, and at different levels. [1] However, with such an overwhelming selection, choosing the right courses that align with a student's academic path, interests, and career aspirations can be a daunting task. Traditional course recommendation systems, which have relied on simple rule-based or manual methods, often fall short of addressing the diverse and evolving needs of modern learners. [5] To solve these challenges, this paper proposes the development of an AI-powered course suggestion system that adapts to the individual needs of each student by using advanced machine learning techniques. Unlike traditional systems, which are confined to static rules or rigid structures, the AI-based system is designed to learn from each student's behaviour, interactions, and academic profile to offer personalized course recommendations that evolve over time.

Keywords: Artificial Intelligence, Machine Learning, Course Recommendation, Personalized Learning, Collaborative Filtering, Content-Based Filtering, Hybrid Recommendation Systems.

I. INTRODUCTION

The rise of online education platforms has significantly transformed the way students access educational content. With platforms such as Coursera, edX, and Udemy offering thousands of courses across various disciplines, students are presented with an overwhelming array of choices. [10] The decision-making process in selecting the most relevant courses has become increasingly complex, as students must align course offerings with their academic goals, personal interests, and career aspirations.

Traditional course selection methods often depend on rigid, rule-based approaches that may fail to address the diverse and ever-changing needs of today's learners. In comparison, more modern approaches offer greater adaptability and relevance., which may be based on manually curated lists or simple rules (e.g., recommend courses based on subject or course level), there is a growing interest in developing more dynamic, adaptive, and personalized course recommendation systems. These systems use data-driven approaches, such as machine learning algorithms, to offer tailored course suggestions, ultimately improving the student learning experience. The challenge, however, lies in the complexity of effectively gathering and utilizing data from multiple sources to create these personalized recommendations.[18]This paper suggests creating an AI-powered course recommendation system that leverages machine learning techniques to deliver personalized, data-driven suggestions for courses.. The system integrates collaborative filtering, content-based filtering, and hybrid models to ensure that course suggestions are relevant, adaptive, and aligned with students' unique learning preferences, academic histories, and career goals. The AI-based system is designed to make course recommendations that continuously evolve as students interact with the platform, thereby providing more accurate and effective suggestions. duals and to the professions they take. [5]Massive open online courses (MOOCs), a form of e-learning, have significantly transformed traditional education and introduced numerous possibilities as an innovative approach. Since their inception, they have enabled individuals from distant locations to access high-quality, engaging, and practical courses offered by prestigious institutions and inspiring instructors at a minimal cost..[7] The increasing variety of courses available in modern educational institutions presents both opportunities and challenges for students. With the vast number of courses to choose from, students often struggle to identify the most suitable ones for their academic and professional goals. Traditional methods of course selection, such as academic advising or course catalogs, may not provide the level of personalization required to optimize the learning journey for each student. In recent years, Artificial Intelligence (AI) has emerged as a promising solution for personalized learning.[19] By utilizing machine learning (ML) algorithms, AI can analyze vast amounts of data to make accurate predictions and provide tailored course suggestions based on student profiles. This paper explores the use of AI in building an intelligent course suggestion system and presents a framework for developing such a system.

2. LITERATURE REVIEW

The need for personalized course recommendations arises due to the distinct nature of each student's academic journey. Students come from varied educational backgrounds, possess different learning styles, and pursue distinct career goals. Therefore, a one-size-fits-all approach to course recommendations is insufficient. Traditional course suggestion systems generally rely on simple rules or heuristic methods, which only provide general suggestions based on broad categories (e.g., recommend courses in a particular discipline or level). While such systems may be effective in some contexts, they are static and fail to capture the dynamic and diverse needs of individual students. Furthermore, these systems often do not account for evolving student interests, learning goals, or preferences. As students complete more courses, their preferences may shift, and they may discover new areas of interest or refine their career objectives. A system that does not adapt to these changes is less likely to provide meaningful course suggestions over time. Additionally, with the rise of vast online course offerings, traditional methods fall short in addressing the complexity of today's educational landscape. Personalized course recommendations powered by AI can address these limitations by learning from students' historical data, preferences, and interactions with the platform. Machine learning provides a powerful tool for improving recommendation systems. By analysing large datasets, machine learning models can uncover hidden patterns in students' behaviour, preferences, and academic progress. Unlike traditional methods, machine learning-based systems can adapt to changing student needs and suggest courses that are more likely to meet their current academic and career aspirations. Recommendation systems powered by machine learning, including collaborative filtering, content-based filtering, and hybrid approaches, are highly effective for this purpose as they can tailor suggestions based on user data.

3. AI-BASED COURSE SUGGESTION PORTFOLIO: ARCHITECTURE AND METHODOLOG

The AI-based course suggestion portfolio is designed to offer personalized course recommendations by utilizing three coreMachine learning methods: collaborative filtering, content-based approaches, and hybrid techniques.. The goal is to create a robust recommendation engine that adapts to the student's academic trajectory, interests, and career goals, making it easier for them to navigate the overwhelming number of available courses.

3.1 Data Collection

The first step in creating personalized course suggestions is gathering relevant data. The system collects data from three primary sources:

1. Student Profiles: This data includes demographic information (e.g., age, location), academic background (e.g., previous courses, grades, skills acquired), and career goals (e.g., desired job roles, industries).[1] It also includes preferences such as learning style (e.g., video-based, text-based learning), time commitment (e.g., full-time, part-time), and areas of interest (e.g., software development, data science).

2. Course Data: This includes detailed information about each course, such as course descriptions, topics covered, prerequisites, skill levels, duration, and instructor profiles. Additionally, course metadata like ratings and reviews from other students helps assess the quality and relevance of courses. [20] The course data also includes whether the course aligns with industry standards, certifications, or relevant career paths.

3. Behavioural Data: This type of data is generated through student interactions on the learning platform. It includes actions such as courses that students have enrolled in, viewed, bookmarked, or rated highly. Behavioural data also includes the amount of time spent on each course, which can indicate a student's engagement level or preference for specific types of content.

3.2 Data Pre-processing

Before the data can be used in machine learning algorithms, it must undergo pre-processing to ensure its quality and accuracy. Key tasks in this phase include:

Normalization: [4] Different features may vary significantly in scale (e.g., ratings on a scale of 1-5, time spent on courses in minutes). Normalization ensures that each feature is on a similar scale, preventing one feature from dominating the model due to its larger range.

Feature Extraction: Important features are extracted from both student profiles and course metadata. For instance, for each course, relevant features could include the subject area, skill level, or instructor, which would be used to build a profile for the course.

Handling Missing Data: [8]I n real-world datasets, missing values are inevitable. Techniques such as imputation (replacing missing values with the mean or median of a feature) or discarding incomplete records are used to handle this issue effectively.

Fig.1 Observation Data

3.3 Collaborative Filtering

Collaborative filtering is a commonly used method that predicts course recommendations by analyzing the preferences and behaviors of users with similar interests. It comes in two types:

User-Based Collaborative Filtering: [17] This method identifies students whose preferences (e.g., courses taken, ratings) are it identifies users with preferences similar to the target student and suggests courses based on what these users have liked or rated highly.

Item-Based Collaborative Filtering: This method analyzes the relationships between courses. It identifies courses that are similar to those the student has already interacted with (e.g., taken or rated highly) and recommends those.

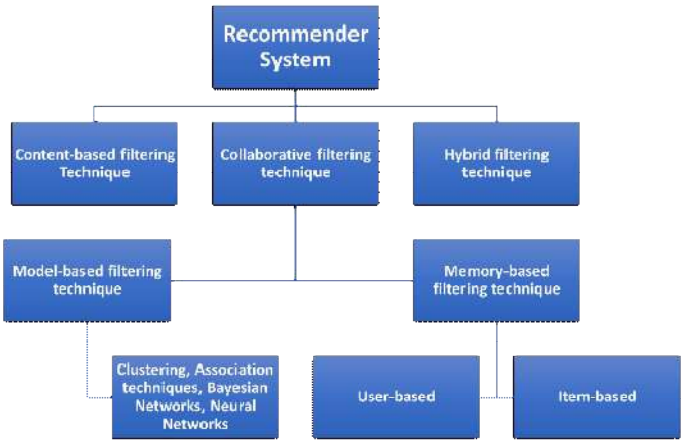

Fig.2 Flowchart

While collaborative filtering is powerful, It encounters the "cold start" issue, particularly when there is a lack of data on new users or newly introduced courses. Without enough interaction history, it becomes difficult to generate reliable recommendations.

3.4 Content-Based Filtering

Content-based filtering recommends courses based on the attributes of the courses themselves. It matches the course metadata (e.g., topics, prerequisites, skills taught) to the student's previous course choices or stated preferences. For example, if a student has previously taken courses on data science, the system may recommend additional courses in related fields, such as machine learning or artificial intelligence. The main advantage of content-based filtering is that it can provide recommendations even when there is little data about other users. However, it has limitations, such as recommending only courses that are very similar to those the student has already interacted with, which may result in a lack of diversity in the recommendations.

3.5 Hybrid Model

Hybrid Recommendation System

A hybrid recommendation system is designed to blendthe advantages of both collaborative and content-based filtering., overcoming their individual limitations to provide more accurate and diverse recommendations.

1. Weighted Averaging

In the weighted averaging approach, predictions from both collaborative and content-based filtering methods are combined. Each method is assigned a weight based on its performance or relevance, and the final recommendation is a weighted average of the predictions from both methods. This approach ensures that the strengths of both methods are utilized.

Collaborative Filtering: This method relies on user interactions and behaviors, such as ratings or clicks, to make recommendations. For example, if many users who liked Course A also liked Course B, the system might recommend Course B to someone who liked Course A.

Content-Based Filtering: This method relies on the attributes of the items themselves, such as the course content, to make recommendations. For instance, if a student has taken courses related to machine learning, the system might recommend other courses in the same field.

Example: Let's say we have two prediction scores for a course recommendation:

- Collaborative filtering score: 80%
- Content-based filtering score: 70%

If we assign weights of 0.6 to collaborative filtering and 0.4 to content-based filtering, the final prediction score is calculated as:

Final Score=(0.6×80%)+(0.4×70%)=48%+28%=76%\text{Final Score} = (0.6 \times 80\%) + (0.4 \times 70\%) = 48\% + 28\% = 76\%

This weighted averaging balances the contribution of both methods, ensuring that neither dominates the recommendation process entirely.

2. Switching

Switching involves dynamically choosing between collaborative and content-based filtering based on the available data or the user's profile. This adaptability ensures that the most suitable method is used in different scenarios.

Cold Start Problem: One of the biggest challenges in recommendation systems is the cold start problem, where the system has little to no data about a new user or item. Content-based filtering can be more effective in this scenario, as it can recommend items based on the attributes of the items themselves, even without user history.

Sufficient Behavioural Data: When the system has accumulated enough data about a user's interactions (e.g., ratings, clicks, purchase history), collaborative filtering becomes more effective. It can leverage the patterns and similarities in user behaviour to provide recommendations.

Example: Imagine a new student joining an online learning platform. Initially, the system lacks sufficient data about the student's preferences. During this phase, content-based filtering can recommend courses based on the student's interests and the content of the courses. As the student interacts with the platform, providing ratings and feedback, the system gradually accumulates behavioural data. Once sufficient data is available, the system can switch to collaborative filtering, utilizing the student's interaction history to refine and personalize the recommendations further.

Benefits of the Hybrid Model

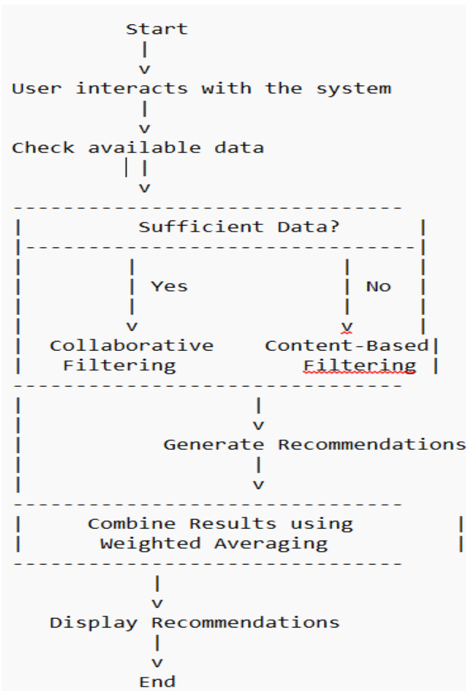

Fig 3 Flowchart for Hybrid Model

Mitigating the Cold Start Problem: By leveraging content-based filtering initially, the hybrid model can provide relevant recommendations even for new users or items. This ensures that users receive useful suggestions from the outset, enhancing their engagement with the platform.

- Enhanced Diversity: Combining collaborative and content-based filtering allows the system to offer a broader range of recommendations. Users can discover courses that align with their previous choices and those popular among similar users, enhancing the diversity and relevance of suggestions.
- Improved Accuracy: The hybrid model's ability to dynamically switch between filtering methods based on data availability ensures that the most effective approach is used at any given time. This adaptability improves the accuracy of recommendations, resulting in higher user satisfaction.

4. SYSTEM IMPLEMENTATION AND EVALUATION

4.1 System Architecture

The system is crafted using Python, a versatile language that excels in both classical machine learning and advanced deep learning applications. This system integrates several key libraries:

Scikit-learn: Used for traditional machine learning models. This library offers straightforward and effective tools for data analysis and modeling.

Tensor Flow: Employed for deep learning-based recommendations. TensorFlow offers robust capabilities for building and deploying machine learning models.

Keras: Acts as an interface for Tensor Flow, simplifying the process of training and evaluating deep learning models.

Key Layers

Data Collection: The system gathers data on student interactions, preferences, and academic history.

Data Preprocessing: The collected data is cleaned and normalized to ensure it is compatible with the machine learning algorithms. This step includes handling missing values, normalizing numerical data, and encoding categorical variables.

Model Training: Using the preprocessed data, the system trains various machine learning models. For deep learning models, Tensor Flow and Keras are utilized.

Evaluation: Models are evaluated using a validation set to ensure they generalize well to new, unseen data. Metrics such as precision, recall, and F1-score are used to assess performance.

Deployment: The system is deployed on a cloud-based infrastructure. This approach ensures that the system is scalable (can handle increasing amounts of data and users) and available (accessible at any time).

User Interface (UI): Web Interface: Students can input their preferences and academic history through an interactive web interface. They can also view the recommended courses and provide feedback on the recommendations.

Feedback Loop: The system continuously incorporates student feedback to refine and improve the recommendation model over time. This iterative process helps in keeping the recommendations relevant and accurate.

```
    +-----------------------+       +-----------------------+       +-----------------------+
|  Data Collection      |-----> |  Data Preprocessing   |-----> |  Model Training       |
+-----------------------+       +-----------------------+       +-----------------------+
                                                                  |       |
                                                              +------+       +------+
                                                              |                     |
                                                      +-------+       +-------------+
                                                      |                     |
                                            +---------+                     +---------+
                                            |                                         |
+-----------------------+       +-----------------------+       +-----------------------+
|  User Interface       |<----->|  Recommendation       |       |  Feedback Loop        |
|  (Web Interface)      |       |  Engine               |       |  (Continuous Refinement)|
+-----------------------+       +-----------------------+       +-----------------------+
```

Fig 4. Flowchart

Training and Validation

Dataset

- Training Set: A large dataset of student-course interactions is used to train the recommendation models.
- Validation Set: A portion of the dataset is set aside to validate the model's performance and ensure it generalizes well to new data.

Evaluation Metrics

- Precision: Measures the accuracy of the recommendations.It is the proportion of relevant recommendations compared to the total number of recommendations provided.
- Recall: Measures the ability to identify all relevant items. It is the ratio of relevant recommendations to the total relevant items available.
- F1-Score: A harmonic mean of precision and recall, providing a single metric that balances both.

The combination of these metrics helps in understanding the effectiveness of the recommendation system from multiple perspectives.

Evaluation Results - Performance

- Precision: The hybrid model achieved a precision of 82%, indicating that 82% of the recommended courses were relevant to the students.

- Recall: A recall of 79% shows that the model successfully identified 79% of all the relevant courses.
- F1-Score: An F1-score of 80% indicates a good balance between precision and recall.

User Feedback

- Student Satisfaction: 90% of students reported that the recommendations significantly improved their course selection process, indicating high user satisfaction and the practical utility of the system.

5. DISCUSSION AND FUTURE WORK

While the AI-based course suggestion system has demonstrated promising results in terms of recommendation accuracy and user satisfaction, several areas can be further refined to enhance its effectiveness and robustness. Let's delve into these aspects in more detail.

Addressing the Cold Start Problem: The cold start problem is a significant challenge in recommendation systems, particularly for new users who have no historical data or for newly introduced courses that lack sufficient interaction data. This issue arises because collaborative filtering relies heavily on past interactions to make accurate recommendations. When such data is sparse or absent, the system struggles to generate meaningful suggestions.

Potential Solutions: **Incorporating Additional Data Sources**: Peer Reviews: Integrating peer reviews and ratings can provide insights into the quality and relevance of courses from the perspective of other students. These reviews can serve as valuable data points, especially for new courses with limited interaction data.

Expert Recommendations: Leveraging recommendations from subject matter experts or course instructors can also fill the data gap. Experts can provide assessments of the course content's quality and its alignment with students' academic goals.

Hybrid Approaches: Combining Collaborative and Content-Based Filtering: By blending these two methods, the system can utilize content-based filtering to make initial recommendations for new users or courses. As more interaction data becomes available, collaborative filtering can take over, refining the recommendations based on user behavior.

Using Contextual Information: Incorporating contextual information such as user demographics, academic background, and learning preferences can help personalize recommendations for new users. This approach ensures that even without historical interaction data, the system can still provide relevant suggestions.

Active Learning and User Feedback: Soliciting Feedback: Actively requesting feedback from users about the recommendations can help the system learn and improve. For instance, asking users to rate the relevance of suggested courses can provide immediate data to enhance future recommendations.

Personalized Questionnaires: Implementing questionnaires that ask new users about their interests, goals, and previous learning experiences can help tailor recommendations. This information serves as an initial profile that the system can use until sufficient interaction data is collected.

Enhancing Recommendation Diversity: While precision and accuracy are crucial, it's also important to ensure that recommendations are diverse. Overly similar recommendations can lead to a narrow learning experience, limiting students' exposure to a broad range of subjects.

Potential Solutions: **Diverse Recommendation Strategies**:

Balancing Novelty and Relevance: Implement strategies that balance recommending familiar courses with introducing novel ones. This approach encourages students to explore new areas while still receiving suggestions aligned with their interests.

Regular Updates: Continuously update the recommendation algorithms to incorporate new courses and emerging topics. This practice ensures that the recommendations remain current and varied.

Implementing Diversity Metrics: **Measuring Diversity**: Introduce metrics to measure the diversity of the recommendations. For example, evaluating the range of topics, difficulty levels, and formats (e.g., lectures, hands-on projects) in the recommended courses. Optimizing for Diversity: Adjust the recommendation algorithms to optimize for diversity, ensuring that the suggestions include a mix of well-known and lesser-known courses.

Improving User Interaction and Engagement: The user interface and overall user experience play a critical role in the effectiveness of the recommendation system. A seamless and engaging interface encourages users to interact with the system, providing valuable feedback and interaction data.

Potential Solutions: **Enhancing the User Interface**:

Intuitive Design: Design an intuitive and user-friendly interface that makes it easy for students to input their preferences, view recommendations, and provide feedback.

Interactive Features: Incorporate interactive features such as course previews, student reviews, and interactive charts to make the user experience more engaging.

Incorporating Gamification Elements: Gamified Feedback: Introduce gamification elements to encourage users to provide feedback. For example, awarding points or badges for rating courses or completing surveys can motivate users to engage more with the system.

Personalized Dashboards: Create personalized dashboards that track students' learning progress, recommended courses, and feedback history. This feature can enhance the sense of personalization and ownership over the learning journey.

FUTURE WORK

To further advance the AI-based course suggestion system, future work could focus on the following areas:

Advanced Natural Language Processing (NLP): Text Analysis: Implement advanced NLP techniques to analyze course descriptions, student reviews, and other textual data. This analysis can provide deeper insights into course content and relevance.

Sentiment Analysis: Use sentiment analysis to gauge the sentiment expressed in student reviews, helping to identify highly recommended courses and potential areas for improvement.

Real-Time Data Integration:

Dynamic Recommendations: Enable the system to provide real-time, dynamic recommendations based on the latest interaction data. This capability ensures that suggestions are always up-to-date and relevant.

Adaptive Learning Models: Implement adaptive Models that continuously adjust and improve by incorporating new data, ensuring the system evolves in line with changing user preferences and trends.

Expanding the Dataset: Collaborative Partnerships: Partner with more educational institutions to expand the dataset, incorporating diverse student demographics and course offerings. A larger and more diverse dataset improves the system's generalizability and robustness.

Cross-Platform Integration: Integrate data from multiple learning platforms to create a comprehensive view of students' learning journeys, enhancing the accuracy and relevance of recommendations.

REFERENCES

1. Othman, M.H.; Mohamad, N.; Barom, M.N. Students' decision making in class selection and enrolment. Int. J. Educ. Manag. 2019, 33, 587–603.
2. Morrow, T. Personalizing Education with Algorithmic Course Selection; Missouri University of Science and Technology: Rolla, MO, USA, 2017.
3. Mourey, J.A.; Markley, M.M.; Koernig, S.K. Dazzling descriptions and tantalizing titles: How simple versus complex course information influences course selection. J. Mark. Educ. 2022, 44, 100–112.
4. Li,Q.;Kim,J. A deep learning-based course recommender system for sustainable development in education. Appl. Sci. 2021, 11, 8993.
5. Kim,T.; Lim, J. Developing an Intelligent Recommendation System for Non-Information and Communications Technology Major University Students. Appl. Sci. 2023, 13, 12774.
6. Wu, B.; Liu, L. Personalized Hybrid Recommendation Algorithm for MOOCs Based on Learners' Dynamic Preferences and Multidimensional Capabilities. Appl. Sci. 2023, 13, 5548.
7. Wu, Z.; Liang, Q.; Zhan, Z. Course Recommendation Based on Enhancement of Meta-Path Embedding in Heterogeneous Graph. Appl. Sci. 2023, 13, 2404.
8. Guruge, D.B.; Kadel, R.; Halder, S.J. The state of the art in methodologies of course recommender systems—A review of recent research. Data 2021, 6, 18.
9. Boratto, L.; Fenu, G.; Marras, M. The effect of algorithmic bias on recommender systems for massive open online courses. In European Conference on Information Retrieval; Springer: Cham, Switzerland, 2019; pp. 457–472.
10. Natarajan, S.; Vairavasundaram, S.; Natarajan, S.; Gandomi, A.H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Syst. Appl. 2020, 149, 113248.
11. Lika, B.; Kolomvatsos, K.; Hadjiefthymiades, S. Facing the cold start problem in recommender systems. Expert Syst. Appl. 2014, 41, 2065–2073.
12. Bang, J.; Lee, B.T.; Park, P. Examination of Ethical Principles for LLM-Based Recommendations in Conversational AI. In Proceedings of the 2023 International Conference on Platform Technology and Service (PlatCon), Busan, Republic of Korea, 16– 18 August 2023; pp. 109–113.
13. Acharya, A.; Singh, B.; Onoe, N. LLM Based Generation of Item-Description for Recommender system. In Proceedings of the 17th ACM Conference on Recommender Systems, Singapore, 18–22 September 2023; pp. 1204–1207.
14. Gao, Y.; Sheng, T.; Xiang, Y.; Xiong, Y.; Wang, H.; Zhang, J. Chat-rec: Towards interactive and explainable LLM-augmented recommender system. arXiv 2023, arXiv:2303.14524.
15. Yin, B.; Xie, J.; Qin, Y.; Ding, Z.; Feng, Z.; Li, X.; Lin, W. Heterogeneous Knowledge Fusion: A Novel Approach for Personalized Recommendation via LLM. In Proceedings of the 17th ACM Conference on Recommender Systems, Singapore, 18–22 September 2023; pp. 599–601.
16. Sanner, S.; Balog, K.; Radlinski, F.; Wedin, B.; Dixon, L. Large Language Model are Competitive Near Cold-start Recommenders for Language-and Item-based Preferences. In Proceedings of the 17th ACM Conference on Recommender Systems, Singapore, 18–22 September 2023; pp. 890–896.
17. Mysore, S.; McCallum, A.; Zamani, H. Large Language Model Augmented Narrative Driven Recommendations. arXiv 2023, arXiv:2306.02250.
18. Rastogi, C.; Tulio Ribeiro, M.; King, N.; Nori, H.; Amershi, S. Supporting human-ai collaboration in auditing LLM with LLM. In Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society, Montréal, QC, Canada, 8–10 August 2023; pp. 913–926.
19. Wu, T.; Terry, M.; Cai, C.J. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 30 April–5 May 2022; pp. 1–22.
20. Holstein, K.; Aleven, V. Designing for human–AI complementarity in K-12 education. AI Mag. 2022, 43, 239–248