Jiang, B. et al. (Eds.) (2025). Proceedings of the 33rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Evaluating the Effectiveness of Large Language Models for Course Recommendation Tasks

Boxuan MA[a*], Md Akib Zabed KHAN[b], Tianyuan YANG[a], Agoritsa POLYZOU[b] & Shin'ichi KONOMI[a]
[a]*Kyushu University, Japan*
[b]*Florida International University, United States*
*boxuan@artsci.kyushu-u.ac.jp

Abstract: Large Language Models (LLMs) have made significant strides in natural language processing and are increasingly being integrated into recommendation systems. However, their potential in educational recommendation systems has yet to be fully explored. This paper investigates the use of LLMs as a general-purpose recommendation model, leveraging their vast knowledge derived from large-scale corpora for course recommendation tasks. We explore prompt and fine-tuning methods for LLM-based course recommendation and compare their performance against traditional recommendation models. Extensive experiments were conducted on a real-world MOOC dataset, evaluating using LLMs as course recommendation systems across a variety of key dimensions. Our results demonstrate that fine-tuned LLMs can achieve good performance comparable to traditional models, highlighting their potential to enhance educational recommendation systems. These findings pave the way for further exploration and development of LLM-based approaches in the context of educational recommendations.

Keywords: Course Recommendation, Large Language Models, MOOC Courses

1. Introduction

Course recommendation systems are increasingly used in the field of education and have become an essential tool in addressing information overload and enhancing user experience for learning (Ma et al., 2020). They can offer personalized course suggestions that align with a student's interests, career goals, or skill development needs and can enhance the educational experience by helping students navigate the vast array of available courses and make more informed decisions about their learning journey (Jiang et al., 2019).

Over the past decade, significant advancements have been made in course recommendation technologies. Traditional recommendation models, including collaborative filtering and content-based methods, have long been employed in practical settings. These approaches typically rely on user-item interaction data or explicit features to provide personalized recommendations (Ma et al., 2021). While successful, traditional recommendation models face notable limitations, such as a lack of generalization and the need for task-specific data for training (Ma et al., 2024). On the other hand, recent deep learning models have demonstrated considerable potential in enhancing prediction accuracy, but they require extensive training and often suffer from a lack of explainability, making them less transparent to users (Dai et al., 2023).

Large Language Models, such as ChatGPT, have gained significant attention due to their great performance for natural language processing tasks, such as text generation, question answering, and language comprehension. These models, with their adaptability and vast knowledge derived from large-scale corpora, present an appealing opportunity for recommendation systems. Previous research has indicated LLMs can be directly used as recommendation systems with prompts, and a growing body of research has begun to explore the potential of LLMs in recommendation tasks, evaluating their performance across various recommendation scenarios and datasets from different domains (Di et al., 2023; Dai et al., 2023; Liu et al., 2023). On the other hand, many researchers have started using LLMs as part of recommendation systems to enhance their performance, such as through feature extraction, feature augmentation, or knowledge representation.

Despite the rapid development of LLMs, most research on LLM-based recommendation systems has focused on domains like music, movies, and books. There has been limited research on applying LLMs specifically to course recommendations within the context of Massive Open Online Courses (MOOCs), and whether LLMs can perform well on course recommendation tasks remains an open question. Therefore, this paper aims to bridge this gap by evaluating the effectiveness of LLMs in recommending courses based on user learning history. Our study offers a comparative analysis between LLMs and traditional recommendation models and investigates the promise of LLMs in addressing key challenges in educational recommendation systems.

2. Related Work

2.1 Course Recommendation

The rise of Massive Open Online Courses (MOOCs) and the increasing number of students have led to the widespread application of course recommendation systems (Ma et al., 2021). Since the introduction of the first course recommendation system based on constraint satisfaction (Parameswaran et al., 2011), various methods have been developed. Content-based approaches recommend courses by matching students' interests with course descriptions and content (Morsomme et al., 2019; Morsy et al., 2019). Matrix Factorization (MF) techniques have also been applied to course recommendation, particularly for predicting future course selections based on students' past courses and grades (Elbadrawy et al., 2015; Sweeney et al., 2016). Other methods have explored the mining of historical course enrollment data to uncover relationships and patterns. For example, Aher et al. (2013) employed association rule mining combined with clustering to identify course relationships, recommending courses based on historical enrollment patterns. Similarly, Bendakir et al. (2006) used association rules with user ratings to enhance the recommendation results, while Polyzou et al. (2019) introduced Scholars Walk, which captures sequential course relationships through a random-walk approach.

As deep learning techniques have gained popularity, they have also been applied to course recommendation systems (Yang et al., 2025; Zhang et al., 2019; Jiang et al., 2019). For instance, Pardos et al. (2019a) modified the skip-gram model to generate course vectors from historical course enrollment data, which are then used to recommend courses similar to a student's previously favored courses. In a similar vein, they proposed the course2vec model, which employs a neural network to generate course recommendations by taking multiple courses as input and predicting a probability distribution over potential course selection (Pardos et al., 2019b).

2.2 LLMs for Recommendation

LLMs have demonstrated their adaptability and significant improvements in a wide range of NLP tasks by leveraging the extensive knowledge from large-scale corpora. Inspired by its successes, there has been a growing interest in applying LLMs to recommendation systems. Recent works have leveraged prompt-based techniques to transform recommendation tasks into natural language tasks, utilizing LLMs without task-specific fine-tuning. For instance, LMRecSys (Zhang et al., 2021) and P5 (Geng et al., 2022) focus on converting recommendation tasks into multi-token cloze tasks using prompts to tackle zero-shot and data efficiency issues. GPT4Rec (Li et al., 2023) and M6-Rec (Cui et al., 2022) utilize LLMs to learn both item and user embeddings. Liu et al. (2023) evaluated ChatGPT's performance on five recommendation scenarios, including rating prediction, sequential recommendation, direct recommendation, explanation generation, and review summarization. Dai et al. (2023) investigated ChatGPT's ranking capabilities, including point-wise, pair-wise, and list-wise ranking. Moreover, the ability of LLMs has also been explored in cold-start scenarios where few user interaction data are available (Wu et al., 2023).

Besides the direct use of LLMs as recommendation systems, LLMs are increasingly being used as components to enhance traditional recommendation models. These approaches integrate LLMs into existing systems through feature extraction, feature augmentation, knowledge representation, and ranking functions (Wu et al., 2024). For instance, Gao et al. (2023) are among the first to use ChatGPT to augment traditional recommender systems by injecting user preferences into the recommendation process through conversational interaction. Another example, Zhang et al. (2023) enhance recommendation system with LLMs by designing prompts for different recommendation settings, where LLM takes candidates from a Recall model for re-ranking.

Despite the growing body of research on LLM-based recommendation systems in different domains, to our knowledge, there has been limited research on applying LLMs specifically to course recommendations aside from work by Khan et al. (2022), while they did not focus on the evaluation of LLMs' potential and only used local models. Yang et al. (2024a, 2024b) use LLMs to generate knowledge concepts from course descriptions and provide course recommendations based on generated concepts. However, whether LLMs can perform well on course recommendation tasks remains an open question.

3. Methods

The workflow for using LLMs in course recommendation tasks is shown in Figure 1. We explore two approaches for applying LLMs to course recommendations. The first approach involves using pre-trained LLMs as recommendation systems, where they generate recommendations directly based on prompts. The second approach involves fine-tuning the model, enriching its knowledge base with student interaction data, and generating recommendations based on the fine-tuned model. The examples of course information, training set, prompt and training instance for fine-tuning can also be seen in Figure 1.

3.1 Direct Use of LLMs for Recommendation

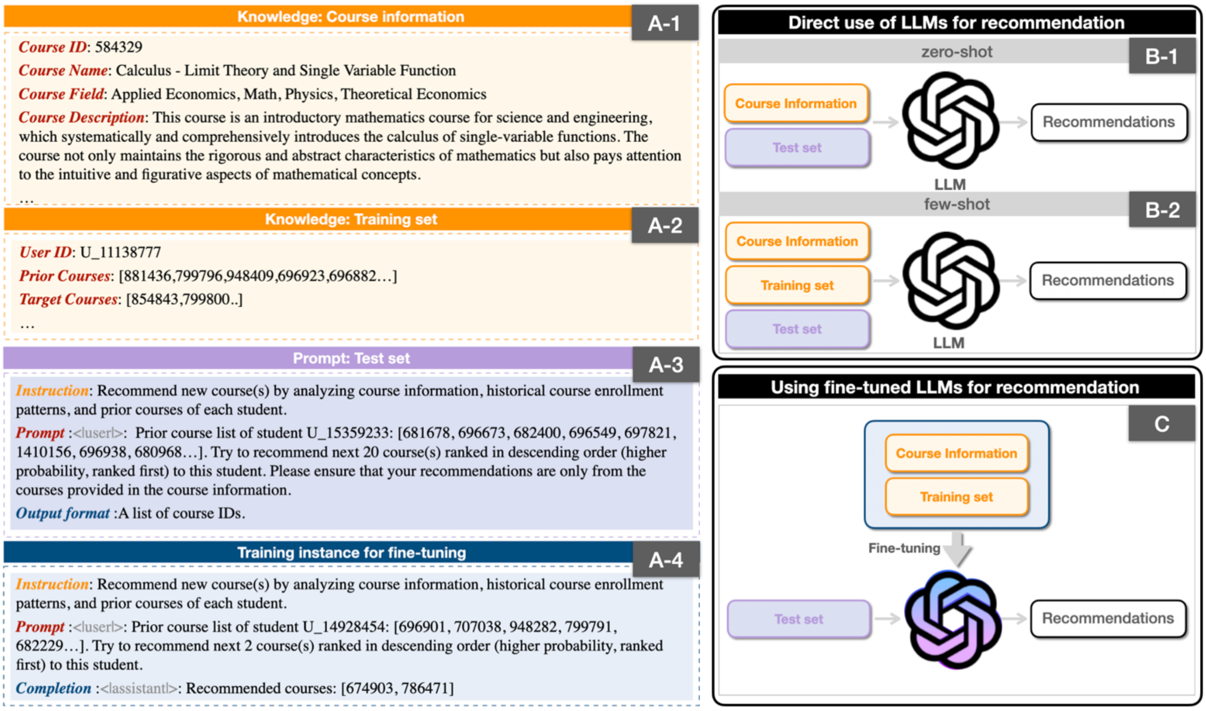
As shown in Figure 1-B, we directly use LLMs to generate recommendations without re-training or fine-tuning the model. Instead, we craft prompts and feed them into the LLMs. The model then generates recommendation results based on the instructions provided in prompts.

For *zero-shot* recommendation (Figure 1-B-1), we provide the LLM with knowledge about all available courses (including course IDs, names, and descriptions), along with the student's prior course registration history as prompt input. The LLM is tasked with recommending a set of courses based on this input. In the case of *few-shot* (Figure 1-B-2), we incorporate additional training data as knowledge context and prompt the LLM to recommend courses based on the complete set of provided prompts. To mitigate hallucination issues and address token limitations, we represent courses using only their IDs in our prompt design.

3.2 Using Fine-tuned LLMs for Recommendation

We also fine-tune LLMs to enhance their knowledge with historical data relevant to our task. We fine-tune two open-source models, Llama-3 (Touvron et al., 2023) and GPT-2 (Radford et al., 2019), as they are freely available and easy to use. Following prior work in item recommendation (Liu et al., 2023), we use students' course enrollment histories to fine-tune the LLMs, enabling them to capture historical enrollment patterns. After fine-tuning, we provide prompts that include a student's prior course registration history and ask the fine-tuned models to recommend a set of courses (see Figure 1-C).

We use <|*user*|> and <|*assistant*|> tokens to indicate the input and output in each training instance, where the input is the student's prior course list, and the output is the subsequent courses they are likely to take. Additionally, we include course descriptions as input to help the model capture the semantic similarity between courses.



Course ID: 584329

Course Name: Calculus - Limit Theory and Single Variable Function

Course Field: Applied Economics, Math, Physics, Theoretical Economics

Course Description: This course is an introductory mathematics course for science and engineering, which systematically and comprehensively introduces the calculus of single-variable functions. The course not only maintains the rigorous and abstract characteristics of mathematics but also pays attention to the intuitive and figurative aspects of mathematical concepts.

…

User ID: U_11138777

Prior Courses: [881436,799796,948409,696923,696882…]

Target Courses: [854843,799800..]

…

Instruction: Recommend new course(s) by analyzing course information, historical course enrollment patterns, and prior courses of each student.

Prompt :<|user|>: Prior course list of student U_15359233: [681678, 696673, 682400, 696549, 697821, 1410156, 696938, 680968…]. Try to recommend next 20 course(s) ranked in descending order (higher probability, ranked first) to this student. Please ensure that your recommendations are only from the courses provided in the course information.

Output format :A list of course IDs.

Instruction: Recommend new course(s) by analyzing course information, historical course enrollment patterns, and prior courses of each student.

Prompt :<|user|>: Prior course list of student U_14928454: [696901, 707038, 948282, 799791, 682229…]. Try to recommend next 2 course(s) ranked in descending order (higher probability, ranked first) to this student.

Completion :<|assistant|>: Recommended courses: [674903, 786471]

Figure 1. Workflow of Using LLMs to Perform Course Recommendation Tasks. A-1: Example Knowledge for Course Information. A-2: Example Knowledge for Training Set. A-3: Example Prompt of Course Recommendation Task (Test Set). A-4: Example of A Training Instance for Fine-Tuning. B-1: Direct Use of LLMs for Recommendation (Zero-Shot). B-2: Direct Use of LLMs for Recommendation (Few-Shot). C: Using Fine-Tuned LLMs for Recommendation.

4. Evaluation

4.1 Dataset

In the context of this work, we focus on the scenario of course recommendation within a MOOC environment. The dataset MOOCCubeX (Yu et al., 2021) in our analysis is collected from the XuetangX[1], one of the largest MOOC websites in China. This dataset consists of 4,216 courses and 3,330,294 students.

4.2 Evaluation Baselines and Metrics

We explore two approaches for using LLMs in course recommendation: direct prompting and fine-tuning with student interaction data. For direct prompting, we use GPT4-turbo and GPT4o due to their popularity and affordability. Following Khan et al. (2022), we fine-tune two open-source models: Llama-3 (Touvron et al., 2023) and GPT2 (Radford et al., 2019).

We also compared LLMs to following traditional baselines: **Random**, recommend random items. **Pop** (Elbadrawy et al., 2016), recommend most popular items. **PMF** (Salakhutdinov et al., 2007), probabilistic matrix factorization relies solely on the user-item interaction matrix. NMF (Lee et al., 1999), non-negative matrix factorization factorizes a non-negative matrix into the product of two or more non-negative matrices based on the user-item interaction matrix. **Item-based KNN** (Sarwar et al., 2001), models user and item based on item similarity obtained by interaction information. **User-based KNN** (Sarwar et al., 2001), models user and item based on user similarity obtained by interaction information. **KEAM** (Yang et al., 2024a), a knowledge graph-based autoencoder model that use both user-item interactions and course concepts.

[1] http://www.xuetangx.com

Table 1. *Accuracy Performance Comparison (%)*

Model	K = 5					K = 10				
	HR	**Recall**	**Precision**	**F1**	**nDCG**	**HR**	**Recall**	**Precision**	**F1**	**nDCG**
Random	0.100	0.005	0.020	0.010	0.020	0.610	0.009	0.080	0.090	0.080
GPT4-turbo zero-shot	0.210	0.100	0.040	0.060	0.050	0.410	0.310	0.040	0.070	0.130
GPT4o zero-shot	0.405	0.080	0.080	0.075	0.075	0.815	0.185	0.080	0.110	0.110
Item-based KNN	0.510	0.070	0.100	0.080	0.125	1.225	0.400	0.130	0.195	0.215
GPT4o few-shot	0.800	0.400	0.160	0.230	0.230	0.800	0.400	0.080	0.130	0.230
GPT4-turbo few-shot	1.002	0.113	0.201	0.147	0.015	1.000	0.110	0.100	0.110	0.100
PMF	1.630	0.285	0.325	0.304	0.435	3.370	0.680	0.340	0.425	0.515
NMF	1.630	0.285	0.325	0.304	0.435	3.370	0.680	0.340	0.425	0.515
User-based KNN	2.960	1.080	0.595	0.755	0.955	4.595	1.645	0.460	0.715	1.100
Pop	8.195	3.300	1.680	2.215	2.680	15.165	5.950	1.660	2.580	3.640
KEAM	13.200	6.053	2.840	3.866	5.977	20.800	9.643	2.300	3.714	7.514
GPT2 Fine-tuning	16.903	7.438	3.524	4.782	11.492	22.643	9.560	2.452	3.903	13.498
Llama3 Fine-tuning	**21.677**	**12.434**	**4.852**	**6.980**	**15.266**	**28.857**	**15.166**	**3.424**	**5.587**	**17.496**

Model	K = 15					K = 20				
	HR	**Recall**	**Precision**	**F1**	**nDCG**	**HR**	**Recall**	**Precision**	**F1**	**nDCG**
Random	1.230	0.150	0.090	0.110	0.110	1.435	0.190	0.070	0.100	0.115
GPT4-turbo zero-shot	0.410	0.310	0.030	0.050	0.130	1.230	0.940	0.060	0.120	0.280
GPT4o zero-shot	1.225	0.450	0.085	0.135	0.185	1.225	0.450	0.060	0.105	0.185
Item-based KNN	2.450	0.590	0.175	0.270	0.320	3.165	0.845	0.175	0.295	0.385
GPT4o few-shot	0.800	0.400	0.500	0.444	0.230	0.800	0.400	0.400	0.400	0.230
GPT4-turbo few-shot	1.000	0.110	0.070	0.080	0.100	3.000	0.118	0.150	0.270	0.370
PMF	5.205	1.465	0.345	0.560	0.885	5.100	0.980	0.260	0.405	0.620
NMF	5.205	1.465	0.345	0.560	0.885	5.100	0.980	0.260	0.405	0.620
User-based KNN	6.530	2.335	0.455	0.760	1.335	8.165	2.830	0.440	0.760	1.530
Pop	17.515	6.655	1.305	2.175	3.865	21.920	8.270	1.295	2.240	4.425
KEAM	26.700	12.229	2.053	3.516	8.667	32.000	14.830	1.915	3.392	9.752
GPT2 Fine-tuning	26.622	10.896	1.990	3.365	14.296	30.793	12.799	1.896	3.303	15.204
Llama3 Fine-tuning	**34.008**	**17.193**	**2.793**	**4.805**	**18.693**	**38.294**	**19.399**	**2.393**	**4.261**	**19.709**

To get a comprehensive evaluation that sheds light on LLMs' performance in the course recommendation task, we select the different metrics following previous works (Dai et al., 2023; Di et al., 2023; Liu et al., 2023). For accuracy metrics, we utilize metrics including ***HR*** (Hit Ratio), ***Recall***, ***Precision***, ***F1***, and ***nDCG*** (normalized Discounted Cumulative Gain). For coverage and novelty metrics, we select ***Coverage*** (quantifies the coverage of all available items that can potentially be recommended), ***Gini Index*** (assesses the distribution of items) and ***EPC*** (Expected Popularity Complement, measures the expected number of relevant recommended items that were not previously seen by the user, showing the model's ability to introduce novelty in the recommendations). Higher scores in these metrics indicate better recommendations.

4.3 Implementation Details

We first split the dataset based on user history, using 80% of the user interaction data for training and 20% for testing. After processing, we randomly sampled 1000 records from the test set for evaluation due to token limitations and expensive costs. For each user, we input their previously interacted items in order and use the LLM to recommend a list of course IDs they might interact with next.

For the pre-trained models, we access GPT4-turbo and GPT4o using OpenAI's API. We also fine-tuned Llama-3-8B and GPT-2-1.5B models. Following previous work (Khan et al., 2022), the Llama-3 model is tokenized using Autotokenizer, and the GPT-2 model is tokenized using the GPT-2tokenizer. We use BitsandBytes[2] and PEFT[3] libraries for fine-tuning efficiently by reducing computing requirements. The PEFT library offers a LoraConfig class to enable QLoRA (Dettmers et al., 2023), which combines quantization with low-rank adapters. By quantizing the base LLM to 4 bits per parameter, we drastically reduce its memory footprint. QLoRA then freezes all original model weights and injects a small set of trainable, low-rank adapter matrices. This approach lets us fine-tune large models using far less GPU memory and compute resources.

[2] https://github.com/bitsandbytes-foundation/bitsandbytes

[3] https://github.com/huggingface/peft

Table 2. *Diversity and Novelty Performance Comparison (%)*

Model	K = 5			K = 10		
	Coverage	**Gini Index**	**EPC**	**Coverage**	**Gini Index**	**EPC**
Pop	0.160	80.000	3.410	0.320	90.000	4.505
PMF	0.220	80.455	0.945	0.395	90.175	1.205
NMF	0.220	80.455	0.945	0.395	90.175	1.205
GPT4-turbo zero-shot	1.140	96.050	0.050	1.910	97.860	0.090
Item-based KNN	6.035	96.375	0.305	9.675	97.775	0.400
User-based KNN	6.270	96.850	3.435	9.295	98.175	1.650
GPT4-turbo few-shot	14.872	99.771	0.250	27.540	99.871	0.250
GPT4o zero-shot	17.375	97.720	0.115	32.910	98.640	0.170
KEAM	12.048	95.560	8.115	18.554	97.466	9.257
Llama3 Fine-tuning	22.505	99.696	**14.399**	31.006	99.692	**15.991**
GPT2 Fine-tuning	23.004	95.896	10.393	25.307	95.699	11.301
GPT4o few-shot	32.400	99.890	0.230	53.750	99.890	0.230
Random	**54.160**	**99.930**	0.090	**78.795**	**99.950**	0.150
	K = 15			K = 20		
Model	**Coverage**	**Gini Index**	**EPC**	**Coverage**	**Gini Index**	**EPC**
Pop	0.480	93.330	4.800	0.640	95.000	5.180
PMF	0.585	93.525	1.350	0.760	95.095	1.335
NMF	0.585	93.525	1.350	0.760	95.095	1.335
GPT4-turbo zero-shot	1.910	97.860	0.090	2.890	98.740	0.130
Item-based KNN	12.485	98.375	0.690	12.485	98.375	0.690
User-based KNN	12.005	98.735	1.825	14.280	99.020	1.955
GPT4-turbo few-shot	37.740	99.900	0.250	47.010	99.900	0.360
GPT4o zero-shot	48.810	98.640	0.170	65.090	99.230	0.200
KEAM	22.988	98.217	9.867	26.747	98.614	10.284
Llama3 Fine-tuning	36.735	99.832	**16.710**	39.802	99.807	**17.198**
GPT2 Fine-tuning	25.894	96.004	11.798	26.092	96.505	12.291
GPT4o few-shot	68.960	99.940	0.230	79.030	99.950	0.230
Random	**90.580**	**99.955**	0.265	**95.900**	**99.960**	0.195

5. Results

5.1 Recommendation Performance

To assess the recommendation capability of large language models (LLMs), we conducted experiments comparing pre-trained and fine-tuned LLMs with traditional models. Table 1 presents the results in percentages.

In summary, we found that the performance of LLMs in the zero-shot prompting setup was relatively low compared to baseline models, outperforming only random recommendation approaches. In contrast, the few-shot prompting setup generally yielded better results, suggesting that providing historical enrollment data helps LLMs identify enrollment patterns and improve recommendation accuracy. However, overall, pre-trained LLMs still fall short of traditional recommendation methods. As a result, relying solely on LLMs for sequential recommendation tasks may not be optimal. Further research is needed to integrate additional guidance and constraints to help LLMs accurately capture historical user interests and produce meaningful recommendations. However, we found that fine-tuned Llama3 consistently outperformed all other methods across different K values, while fine-tuned GPT-2 surpassed traditional baselines and achieved performance similar to KEAM, which is the state-of-the-art model. These results suggest that fine-tuning enables LLMs to specialize in the recommendation task, allowing them to more effectively model user behavior and course relationships and thereby produce more accurate predictions.

5.2 Diversity and Novelty Performance

We also aim to assess the extent of diversity (**Coverage, Gini Index**) and novelty (**EPC**) in the recommendations generated by LLMs. Table 2 presents the results in percentages.

Overall, the Random model achieves the highest Coverage and Gini Index across all K values, outperforming all other models. This result is not surprising, given its random nature, which ensures a broad range of items are recommended. In contrast, LLMs perform relatively better in diversity and novelty dimensions. Notably, the advanced model, GPT4o, outperforms

Table 3. *Performance Comparison (%) on Cold Start Scenario*

	K = 5							
Model	**HR**	**Recall**	**Precision**	**F1**	**nDCG**	**Coverage**	**Gini Index**	**EPC**
Random	0.000	0.000	0.000	0.000	0.000	**53.080**	**99.930**	0.000
PMF	0.000	0.000	0.000	0.000	0.000	0.190	80.190	0.000
NMF	0.000	0.000	0.000	0.000	0.000	0.190	80.190	0.000
GPT4-turbo	0.200	0.200	0.040	0.070	0.090	33.290	98.710	0.050
Item-based KNN	0.430	0.430	0.090	0.140	0.220	11.660	97.280	0.150
User-based KNN	0.430	0.430	0.090	0.140	0.430	11.280	98.650	0.430
GPT4o	1.080	1.080	0.220	0.360	0.820	23.630	98.710	0.740
Pop	4.730	4.730	0.950	1.580	3.030	0.160	80.000	2.470
GPT2 Fine-tuning	4.409	4.409	0.882	1.470	3.030	9.037	83.844	1.920
KEAM	7.800	7.800	1.560	2.600	6.406	34.294	97.108	4.933
Llama3 Fine-tuning	**13.613**	**13.613**	**2.723**	**4.538**	**11.810**	7.795	83.029	**11.473**
	K = 10							
Model	**HR**	**Recall**	**Precision**	**F1**	**nDCG**	**Coverage**	**Gini Index**	**EPC**
Random	0.220	0.220	0.020	0.040	0.060	**77.570**	**99.950**	0.020
PMF	0.000	0.000	0.000	0.000	0.000	0.350	90.100	0.000
NMF	0.000	0.000	0.000	0.000	0.000	0.350	90.100	0.000
GPT4-turbo	0.200	0.200	0.200	0.400	0.090	61.750	99.240	0.050
Item-based KNN	1.080	1.080	0.110	0.200	0.650	15.410	99.130	0.520
User-based KNN	1.720	1.720	0.170	0.310	0.650	18.460	98.310	0.310
GPT4o	1.510	1.510	0.150	0.270	0.960	41.520	99.240	0.800
Pop	6.670	6.670	0.670	1.210	3.640	0.320	90.000	2.710
GPT2 Fine-tuning	7.214	7.214	0.721	1.312	3.420	9.156	90.953	2.266
KEAM	11.000	11.000	1.100	2.000	7.945	56.292	98.484	5.346
Llama3 Fine-tuning	**16.515**	**16.515**	**1.651**	**3.003**	**12.632**	7.795	90.631	**11.904**

GPT4-turbo model across all dimensions. Upon a closer examination of the generated recommendations, we observed that GPT4-turbo often exhibits “lazy behaviors”, generating similar or repetitive recommendation lists, which leads to low diversity. Furthermore, few-shot models consistently outperform zero-shot models. By learning from user-specific data, few-shot models can generate more personalized and relevant recommendations, enhancing both diversity and novelty. Finally, fine-tuned models like GPT-2 and Llama3 not only excel in diversity but also significantly surpass other models in novelty. This indicates that directly applying LLMs to recommendation tasks is challenging because the data used for pre-training LLMs differs from the specific requirements of recommendation tasks. However, fine-tuned LLMs, when trained on specific data, can deliver better results.

5.3 Cold Start Scenario

Cold start is a well-known challenge in course recommendation systems, especially in MOOC environments. It refers to the difficulty of recommending relevant courses to new users who lack sufficient interaction data. To investigate the performance of LLMs in cold start scenarios for course recommendations, we adopt two different experiments inspired by previous studies (Di et al., 2023; Dai et al., 2023).

First, we identify cold-start users by dividing the users of the dataset into quartiles based on their historical interaction data. The lower quartile, representing users with the least interaction, is selected as the subset of cold-start users. This allows us to evaluate the models under consistent cold-start conditions, ensuring that all models are tested with a similar subset of users (note that we fine-tuned our LLMs using the same cold-start training set). The results of this evaluation are presented in Table 3. We observe that off-the-shelf LLMs outperform traditional models for cold start scenarios when only limited training data is available. LLMs do not require extensive training data to function as recommendation systems, as their pre-trained knowledge allows them to make informed predictions. The Pop method, by contrast, performs well because it simply recommends the most popular courses in the dataset. Moreover, with minimal training data, fine-tuned Llama3 achieved the highest scores, while fine-tuned GPT-2 outperformed Pop and ranked second only to KEAM. This demonstrates that the reasoning abilities and extensive knowledge encoded in LLMs enable them to generate superior recommendations.

Secondly, we investigate the amount of training data required for traditional recommendation models to achieve performance comparable to or better than LLMs.

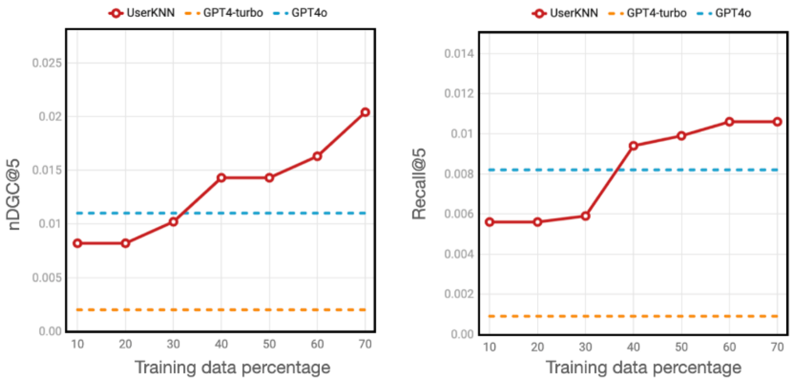
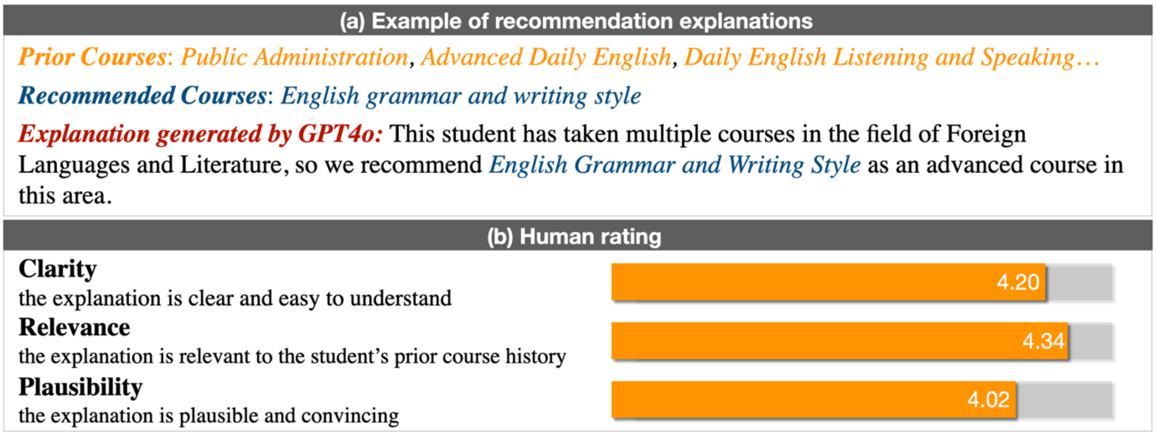


Figure 2. Comparison with UserKNN Using Different Percentages of Training Data.



(a) Example of recommendation explanations

Prior Courses: *Public Administration*, *Advanced Daily English*, *Daily English Listening and Speaking…*

Recommended Courses: *English grammar and writing style*

Explanation generated by GPT4o: This student has taken multiple courses in the field of Foreign Languages and Literature, so we recommend *English Grammar and Writing Style* as an advanced course in this area.

(b) Human rating

Clarity
the explanation is clear and easy to understand

Relevance
the explanation is relevant to the student's prior course history

Plausibility
the explanation is plausible and convincing

Figure 3. (a) Example of Explanation Generated by LLMs (b)Human Rating Results.

Specifically, we chose the User-based KNN model as it performed well in the first experiment. We then evaluated their performance after training on varying proportions of training data and compared their performance to that of LLMs. *Recall@5* and *nDCG@5* are reported in Figure 2. As expected, the performance of User-based KNN improves with increasing amounts of training data. Also, we can observe that although GPT4-turbo's performance is not good, direct use of GPT4o as a recommendation system without training data still outperforms User-based KNN that trained on few data, i.e., less than 30%. Based on these findings, we conclude that using LLMs as course recommendation systems is a promising approach for mitigating the cold-start problem, offering effective solutions when traditional methods may struggle.

5.4 Explanation

Providing students with clear and understandable explanations for course recommendations can enhance the transparency and perceived trustworthiness of the system, thereby improving overall user satisfaction (Ma et al., 2021; Ma et al., 2024). We also prompt LLMs to produce a textual explanation for each recommended course.

To assess the quality of these explanations, we conducted a small-scale human evaluation on 50 sampled outputs. Each explanation was rated along three dimensions, including ***Clarity*** (the explanation is clear and easy to understand), ***Relevance*** (the explanation is relevant to the student's prior course history), and ***Plausibility*** (the explanation is plausible and convincing), using a 5-point Likert scale by two human researchers, the example explanation and human rate results are shown in Figure 3. Most explanations received scores of 4 or higher across all dimensions, suggesting that the explanations were generally interpretable and credible. These findings indicate that LLMs can effectively generate intuitive and trustworthy justifications by contextualizing recommendations within students past learning experiences.

6. Conclusions

In this paper, we evaluate the performance of LLMs in course recommendation tasks and compare them with traditional recommendation methods across diverse settings. Our results indicate that LLMs without fine-tuning perform worse than baseline models. However, fine-tuned LLMs significantly outperform traditional approaches, particularly in cold-start scenarios. Moreover, LLMs demonstrate strong capabilities in generating high-quality recommendation explanations. These findings provide valuable insights into the strengths and limitations of LLMs in educational recommendation systems.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP24K20903 and Dissertation Fellowship awarded by the University Graduate School of Florida International University.

References

Aher, S.B., Lobo, L.: Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data. *Knowledge-Based Systems* 51, 1–14 (2013)

Bendakir, N., Aïmeur, E.: Using association rules for course recommendation. In: *Proceedings of the AAAI workshop on educational data mining*. vol. 3, pp. 1–10 (2006)

Cui, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint* arXiv:2205.08084 (2022)

Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., Sun, Z., Zhang, X., Xu, J.: Uncovering chatgpt's capabilities in recommender systems. In: *Proceedings of the 17th ACM Conference on Recommender Systems*. pp. 1126–1132 (2023)

Di Palma, D., Biancofiore, G.M., Anelli, V.W., Narducci, F., Di Noia, T., Di Sciascio, E.: Evaluating chatgpt as a recommender system: A rigorous approach. *arXiv preprint* arXiv:2309.03613 (2023)

Elbadrawy, A., Karypis, G.: Domain-aware grade prediction and top-n course recommendation. In: *Proceedings of the 10th ACM conference on recommender systems*. pp. 183–190 (2016)

Elbadrawy, A., Studham, R.S., Karypis, G.: Collaborative multi-regression models for predicting students' performance in course activities. In: *proceedings of the fifth international conference on learning analytics and knowledge*. pp. 103–107 (2015)

Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., Zhang, J.: Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint* arXiv:2303.14524 (2023)

Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y.: Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In: *Proceedings of the 16th ACM Conference on Recommender Systems*. pp. 299–315 (2022)

Jiang, W., Pardos, Z.A., Wei, Q.: Goal-based course recommendation. In: *Proceedings of the 9th international conference on learning analytics & knowledge*. pp. 36–45 (2019)

Khan, M.A.Z., Polyzou, A., Bennamane, N.: How can we use llms for edm tasks? the case of course recommendation (2022)

Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–91 (11 1999). https://doi.org/10.1038/44565

Li, J., Zhang, W., Wang, T., Xiong, G., Lu, A., Medioni, G.: Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint* arXiv:2304.03879 (2023)

Liu, J., Liu, C., Zhou, P., Lv, R., Zhou, K., Zhang, Y.: Is chatgpt a good recommender? a preliminary study. *arXiv preprint* arXiv:2304.10149 (2023)

Liu, J., Liu, C., Zhou, P., Ye, Q., Chong, D., Zhou, K., Xie, Y., Cao, Y., Wang, S., You, C., et al.: Llmrec: Benchmarking large language models on recommendation task. *arXiv preprint* arXiv:2308.12241 (2023)

Ma, B., Lu, M., Taniguchi, Y., Konomi, S.: Exploration and explanation: An interactive course recommendation system for university environments. In: *IUI Workshops* (2021)

Ma, B., Lu, M., Taniguchi, Y., Konomi, S.: CourseQ: the impact of visual and interactive course recommendation in university environments. *Research and practice in technology enhanced learning* 16, 1–24 (2021)

Ma, B., Lu, M., Taniguchi, Y., Konomi, S.: Investigating course choice motivations in university environments. *Smart Learning Environments* 8(1), 1–18 (2021)

Ma, B., Taniguchi, Y., Konomi, S.: Course recommendation for university environments. *International educational data mining society* (2020)

Ma, B., Yang, T., Ren, B.: A survey on explainable course recommendation systems. In: Streitz, N.A., Konomi, S. (eds.) *Distributed, Ambient and Pervasive Interactions*. pp. 273–287. Springer Nature Switzerland, Cham (2024)

Morsomme, R., Alferez, S.V.: Content-based course recommender system for liberal arts education. *International educational data mining society* (2019)

Morsy, S., Karypis, G.: Will this course increase or decrease your gpa? towards grade-aware course recommendation. *Journal of Educational Data Mining* 11(2), 20–46 (2019)

Parameswaran, A., Venetis, P., Garcia-Molina, H.: Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems* (TOIS) 29(4), 1–33 (2011)

Pardos, Z.A., Fan, Z., Jiang, W.: Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User modeling and user-adapted interaction* 29, 487–525 (2019a)

Pardos, Z.A., Jiang, W.: Combating the filter bubble: Designing for serendipity in a university course recommendation system. *arXiv preprint* arXiv:1907.01591 (2019b)

Polyzou, A., Nikolakopoulos, A.N., Karypis, G.: Scholars walk: A markov chain framework for course recommendation. *International Educational Data Mining Society* (2019)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* 1(8), 9 (2019)

Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. pp. 1257–1264 (2007)

Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*. pp. 285–295 (2001)

Sweeney, M., Rangwala, H., Lester, J., Johri, A.: Next-term student performance prediction: A recommender systems approach. In: *arXiv preprint* arXiv:1604.01840 (2016)

Yang, T., Ren, B., Ma B., He, T., Gu, C., Konomi, S.: Boosting course recommendation explainability: A knowledge entity aware model using deep learning. In: *International Conference on Computers in Education* (2024a)

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. *arXiv preprint* arXiv:2302.13971 (2023)

Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., et al.: A survey on large language models for recommendation. *World Wide Web* 27(5), 60 (2024)

Wu, X., Zhou, H., Yao, W., Huang, X., Liu, N.: Towards personalized cold-start recommendation with prompts. *arXiv preprint* arXiv:2306.17256 (2023)

Yang, T., Ren, B., Gu, C., Ma, B., Konomi, S.: Leveraging ChatGPT for automated knowledge concept generation. In: *21st International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA* 2024. pp. 75–82. IADIS Press (2024b)

Yu, J., Wang, Y., Zhong, Q., Luo, G., Mao, Y., Sun, K., Feng, W., Xu, W., Cao, S., Zeng, K., et al.: MOOCCubeX: A large knowledge-centered repository for adaptive learning in MOOCs. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. pp. 4643–4652 (2021)

Zhang, J., Hao, B., Chen, B., Li, C., Chen, H., Sun, J.: Hierarchical reinforcement learning for course recommendation in moocs. In: *Proc. AAAI conf artificial intelligence*. vol. 33(1), pp. 435–442 (2019)

Zhang, J., Xie, R., Hou, Y., Zhao, X., Lin, L., Wen, J.R.: Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems* (2023)

Zhang, Y., DING, H., Shui, Z., Ma, Y., Zou, J., Deoras, A., Wang, H.: Language models as recommender systems: Evaluations and limitations. In: I (Still) Can't Believe It's Not Better! *NeurIPS* 2021 Workshop (2021)

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, *36*, 10088-10115. (2023)

Frej, J., Shah, N., Knezevic, M., Nazaretsky, T., & Käser, T. Finding paths for explainable mooc recommendation: A learner perspective. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*. pp. 426-437 (2024).

Yang, T., Ren, B., Gu, C., Ma, B., He, T., & Konomi, S. I. Towards better course recommendations: integrating multi-perspective meta-paths and knowledge graphs. In *Proceedings of the 15th international learning analytics and knowledge conference*. pp. 137-147 (2025).