Taxonomy-Guided Zero-Shot Recommendations with LLMs

Yueqing Liang[1], Liangwei Yang[2], Chen Wang[2], Xiongxiao Xu[1], Philip S. Yu[2], Kai Shu[3]*

[1]Illinois Institute of Technology, Chicago, IL
[2]University of Illinois at Chicago, Chicago, IL
[3]Emory University, Atlanta, GA
{yliang40, xxu85}@hawk.iit.edu
{lyang84, cwang266, psyu}@uic.edu
kai.shu@emory.edu

Abstract

With the emergence of large language models (LLMs) and their ability to perform a variety of tasks, their application in recommender systems (RecSys) has shown promise. However, we are facing significant challenges when deploying LLMs into RecSys, such as limited prompt length, unstructured item information, and un-constrained generation of recommendations, leading to sub-optimal performance. To address these issues, we propose a novel Taxonomy-guided Recommendation (TAXREC) framework to empower LLM with category information in a systematic approach. Specifically, TAXREC features a two-step process: one-time taxonomy categorization and LLM-based recommendation. In the one-time taxonomy categorization phase, we organize and categorize items, ensuring clarity and structure of item information. In the LLM-based recommendation phase, we feed the structured items into LLM prompts, achieving efficient token utilization and controlled feature generation. This enables more accurate, contextually relevant, and zero-shot recommendations without the need for domain-specific fine-tuning. Experimental results demonstrate that TAXREC significantly enhances recommendation quality compared to traditional zero-shot approaches, showcasing its efficacy as a personal recommender with LLMs. Code is available at: https://github.com/yueqingliang1/TaxRec.

1 Introduction

Due to the emergent ability (Wei et al., 2022), large language models (LLMs) have triggered the purse of artificial general intelligence (AGI) (Fei et al., 2022), where an artificial intelligence (AI) system can solve numerous tasks. Tasks that were previously completed separately are now combined into one language modeling task by using prompt templates to turn them into sentences. As shown in Figure 1(a), one single LLM (Achiam et al., 2023) can act as our personal assistant to complete a series of tasks such as question answering (Tan et al., 2023), machine translation (Zhang et al., 2023a) and grammar checking (Yasunaga et al., 2021). Besides, an LLM-based assistant can also provide reasonable recommendations with its own knowledge within the pre-trained parameters (Gao et al., 2023). Without the need for fine-tuning on historical user-item interactions, it acts as the zero-shot recommenders, which greatly extends LLMs toward a more generalized all-task-in-one AI assistant.

Acting as the assistant for recommendation, LLMs face several challenges when it meets the requirement from recommender system (RecSys) as shown in Figure 1(b). (1) Limited prompt length prohibits input of all items. In RecSys, the size of item pool effortlessly grows over millions with each represented by tens of tokens, which easily surpasses the prompt length limit (Pal et al., 2023) of LLMs. Let alone the long context also causes decoding problems (Liu et al., 2024) even the whole item pool is small enough to fit within the prompt. (2) Vague and unstructured item title and description. The text information of items is provided at the will of merchant, which is usually unstructured and vague (Ni et al., 2019) to understand without sufficient contexts. As shown in Figure 1(b), the title "1984" can represent the year/book/movie and "Emma" is able to represent people name/book. Direct recommendation with the raw item titles can suffer from the ambiguity prompt issue and leads to inferior performance. (3) Un-constrained generation out of candidate item pools. The generation process of LLMs is un-constrained, and can easily be un-matchable within the item pool, especially for the unstructured titles. For example, the LLMs can generate an item "Punch-Out!!!" that totally out of the item pool when we only provide user's historical interactions. With the direct text-based

*Corresponding author

Proceedings of the 31st International Conference on Computational Linguistics, pages 1520–1530
January 19–24, 2025.

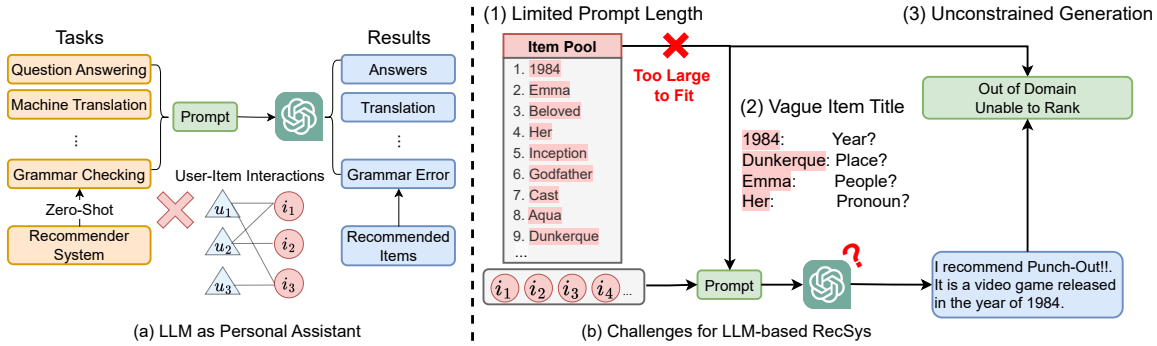


Figure 1: (a) LLMs are zero-shot recommenders without knowing other user-item interactions when acting as personal recommendation assistants. (b) Three challenges occur when integrating RecSys within the all-task-in-one LLM assistant, i.e., limited prompt length, vague item title and unconstrained generation.

generation, it is also compute intensive and mostly infeasible to calculate the ranking score for all candidate items within the pool.

In this paper, we propose leveraging a taxonomy dictionary to address the aforementioned challenges. A taxonomy provides a systematic framework for identifying, organizing, and grouping items effectively. For each dataset, we first retrieve an LLM to obtain a taxonomy dictionary that contains the categorization knowledge of a domain. This then enables us to categorize all candidate items into a structured item pool, thereby mitigating the issue of vague item titles or descriptions by providing richer item contextual information. For instance, an item like "1984" can be clarified as "Type: Book, Genre: Fiction, Theme: Power, . . . ". When prompting the LLM for recommendations, we incorporate the taxonomy dictionary into the prompt to enrich the model's understanding of the candidate items and their attributes.

The taxonomy dictionary is a condensed categorization of the whole item pool. Compared with adding all candidate items, adding the dictionary can greatly save the tokens needed to inform LLM the candidates information, alleviating the limited prompt length challenge. Instead of directly generating tokens within the item title, we propose to generate categorized features from the taxonomy dictionary. As the taxonomy dictionary can be easily fed within the prompt, it is more controllable to generate features within the dictionary with our designed prompt template. We finally calculate the feature matching score within the categorized item pool to rank the items for recommendation.

Our taxonomy-based approach is a two-step process. The first is a one-time taxonomy categorization step, which retrieves knowledge from LLM to build a taxonomy and a categorized item pool. The second is an LLM-based Recommendation step, which infers user's preference based on their historical interactions. This approach effectively handles large item pools, making it feasible to work within LLM token limits, leading to a more efficient, accurate, and scalable recommendation process. Our contributions are summarized as:

- The development of a systematic taxonomy dictionary framework to categorize and organize items, enhancing the structure and clarity of item information.
- We propose TAXREC, a taxonomy-based method to retrieve knowledge and enhance LLM's ability as personal recommender.
- Experiments show significant improvement of TAXREC over current zero-shot recommenders, proving the effectiveness of our proposed item taxonomy categorization.

2 Related Work

2.1 LLM for Recommendations

Recommendation systems are crucial for helping users discover relevant and personalized items (Wang et al., 2024a,b; Yang et al., 2024). With the rise of LLMs, there has been growing interest in utilizing these models to improve recommendation systems (Cao et al., 2024; Wang et al., 2024c). LLM-based recommenders can be broadly divided into two categories: discriminative and generative (Wu et al., 2023). Discriminative methods use LLMs to learn better user and item representations from contextual information (Hou et al., 2022;

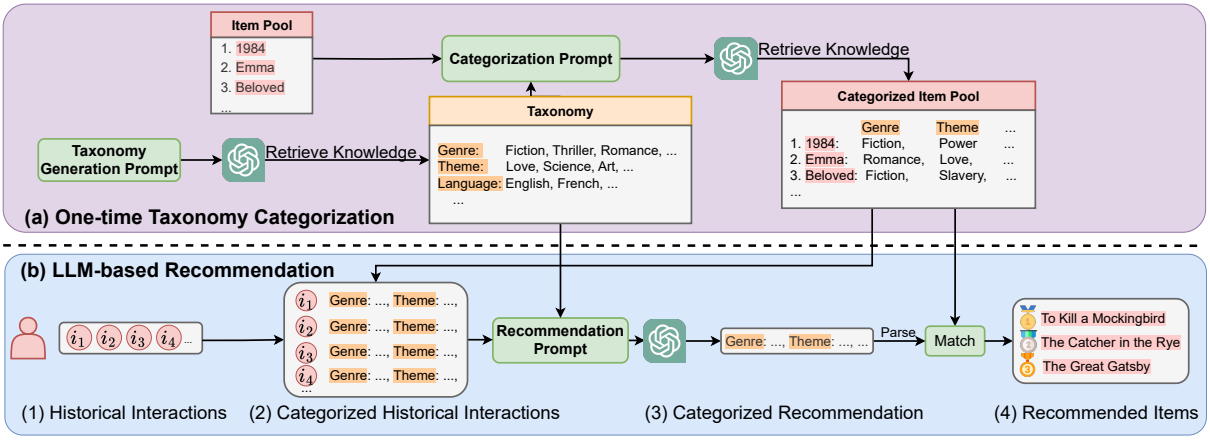


Figure 2: The proposed **TAXREC** for zero-shot LLM-based recommendation. (a) One-time Taxonomy Categorization step aims to generate in-domain taxonomy and enrich/categorize item's title into structured text information. (b) LLM-based Recommendation step provides ranked item lists for users based on the user's historical interactions.

Li et al., 2023; Xiao et al., 2022; Zhang et al., 2022; Yuan et al., 2023; Yao et al., 2022), while generative methods leverage LLMs' ability to generate recommendations by framing traditional ranking tasks as natural language tasks. Instead of computing scores, generative systems use techniques like prompt tuning and in-context learning to produce recommendations directly.

One of the first generative approaches was (Geng et al., 2022), which utilized the pre-trained T5 model. More recent works have explored using LLMs for recommendations without fine-tuning (Dai et al., 2023; Gao et al., 2023; Liu et al., 2023; Lyu et al., 2023; Wang et al., 2023c). (Wang et al., 2023b) integrated databases with LLMs to act as autonomous agents for recommendation tasks, and (Wang et al., 2023a) augmented user-item interaction graphs with LLMs. (Lyu et al., 2023) studied the impact of different prompts on recommendation outcomes, while (Wang et al., 2023c) introduced a multi-round self-reflection framework for sequential recommendations.

However, these works did not address the challenge of vague and unstructured text representations of items, which can hinder LLMs' recommendation capabilities. In this paper, we propose a taxonomy-guided framework to address this issue.

2.2 Zero-shot Recommendations

Zero-shot recommendation has become an important area in recommendation systems, focusing on predicting user preferences without parameter adjustment. Various approaches have been developed to tackle this challenge, such as content-based methods that use item attributes (Lian et al., 2018; Zhang et al., 2023b; Cao et al., 2023) and techniques that leverage pre-trained language models to extract item and user characteristics from text (Ding et al., 2022; Hou et al., 2022; Li et al., 2023). Recently, LLMs have been explored for zero-shot recommendations (Hou et al., 2024; Wang and Lim, 2023; He et al., 2023; Feng et al., 2024; Wang et al., 2023b). However, these studies often face limitations due to the restricted context length of LLMs, making it difficult to input all items. Some approaches address this by incorporating external tools (Wang et al., 2023b; Feng et al., 2024), while others use a plug-in recommendation model to narrow the candidate pool (Hou et al., 2024; Wang and Lim, 2023; He et al., 2023). Despite these efforts, none have fully solved this problem using LLM knowledge alone. Our work introduces a taxonomy-guided LLM recommender that compresses the item pool via an LLM-generated taxonomy, enabling effective zero-shot recommendations without external knowledge.

3 Methodology

In this paper, we aim to use LLMs as zero-shot recommenders. To achieve this, we propose a framework TAXREC that uses taxonomy as an intermediate to retrieve the knowledge in LLMs. Specifically, our TAXREC contains two phases. The first one is a one-time taxonomy categorization phase, and the second one is the LLM-based recommendation phase. The overall framework of TAXREC is shown in Figure 2. Next, we will introduce the two phases of our proposed framework TAXREC in detail.

3.1 Problem Formulation

Without other user-item interactions, LLMs act as zero-shot recommenders when users directly seek recommendations. The task is to generate the Top-k recommended items is from the candidate item pool $\mathcal{I} = \{i_j\}_{j=1}^{|\mathcal{I}|}$ only based on user's historical interactions $\mathcal{H} = \{i_1, i_2, ..., i_{|\mathcal{H}|}\}$ and the knowledge within LLMs. As a pure text-based approach, each item i is a title string as shown in Figure 2. The task can then be represented as designing an LLM-based function:

$$i_1, i_2, ..., i_k = f_{LLM}(\mathcal{H}). \quad (1)$$

In TAXREC, we further propose a taxonomy dictionary $\mathcal{T}$ as an intermediate to better retrieve knowledge from LLMs, as well as a categorized item pool $\mathcal{I}^{\mathcal{C}} = \{i_j^C\}_{j=1}^{\mathcal{I}^{\mathcal{C}}}$ and categorized historical interactions $\mathcal{H}^{\mathcal{C}} = \{i_1^C, i_2^C, ..., i_{\mathcal{H}^{\mathcal{C}}}^C\}$.

3.2 One-time Taxonomy Categorization

The first step is a one-time generation, which aims to structure and clarify items into a categorized item pool. The original item text representation is vague and unstructured, which poses challenges for LLMs to understand and infer user's interest. As the first item within the pool shown in Figure 2(a), "1984" can be represented as either year/book/movie. Without sufficient in-domain background knowledge, direct recommendation in zero-shot manner with these vague and unstructured textual information is challenging for LLMs.

To make LLMs better understand the key information in the historical interactions, we first extract the in-domain taxonomy dictionary from LLMs with a designed taxonomy generation prompt:

$$\mathcal{T} = f_{LLM}(P_{\text{Taxonomy\_Gen}}), \quad (2)$$

where $P_{\text{Taxonomy\_Gen}}$ is the Taxonomy Generation Prompt as shown in Table 1. It is designed to retrieve the in-domain knowledge from LLM to better classify items. As shown in Figure 2, we can obtain the important attributes to classify books such as Genre, Theme, Language, etc. With a well-defined taxonomy dictionary $\mathcal{T}$, we are able to enrich and categorize each item i as:

$$i^C = f_{LLM}(P_{\text{Categorization}}|i, \mathcal{T}), \quad (3)$$

where $P_{\text{Categorization}}$ is the Categorization Prompt as shown in Table 1 to obtain i's categorized feature list as $i^C = [f_1, f_2, ..., f_{|i^C|}]$. We can structure and enrich item textual descriptions with knowledge from LLMs. For example, as shown in the categorized item pool in Figure 2(a), the book "1984" is enriched with "fiction" as genre and "power" as theme. Compared with the original vague book title, the enriched texts provide more detailed information to assist LLMs inference user's interests. The categorized item pool $\mathcal{I}^C$ is obtained by categorizing items in $\mathcal{I}$ with Equation 3.

Though we infer LLMs two times in this step, this is a one-time operation for the current domain, and the results could be stored for next step usage.

Table 1: Examples of the three prompts in our proposed TAXREC for book recommendations.

	Prompts
Taxonomy Generation Prompt	You are an expert in book recommendations. I have a book dataset. Generate a taxonomy for this book dataset in JSON format. This taxonomy includes some features, each with several values. It is used for a book recommendation system.
Categorization Prompt	You are a book classifier. Given a book, please classify it following the format of the given taxonomy. <Taxonomy $\mathcal{T}$> <Book i>
Recommendation Prompt	You are a book recommender system. Given a list of books the user has read before, please recommend k books in a list of features following the format of the given taxonomy. <Taxonomy $\mathcal{T}$> <Categorized historical interactions $\mathcal{H}^{\mathcal{C}}$>

3.3 LLM-based Recommendation

In the second step, we take the advantage of $\mathcal{I}^C$ and $\mathcal{T}$ generated in Section 3.2, and build an LLM-based recommender for the user. The process is shown in Figure 2(b). We first process each user's historical interactions $\mathcal{H}$ to categorized historical interactions $\mathcal{H}^{\mathcal{C}}$ by mapping item from $\mathcal{I}^C$. In this way, the item representation will be structured and enriched based on the taxonomy. We then combine $\mathcal{H}^{\mathcal{C}}$ with taxonomy $\mathcal{T}$ to form a prompt to obtain the categorized recommendation as:

$$s = f_{LLM}(P_{\text{Recommendation}}|\mathcal{H}^{\mathcal{C}}, \mathcal{T}), \quad (4)$$

where s is the categorized recommendation, which is a text sequence of key-value pairs representing

item's features within $\mathcal{T}$. $P_{\text{Recommendation}}$ is the recommendation prompt given $\mathcal{H}^{\mathcal{C}}$ and $\mathcal{T}$as shown in Table 1. Using $\mathcal{T}$ instead of the item pool can greatly decrease the prompt length and fit the in-domain item's information within the limited context requirement from LLMs. $P_{\text{Recommendation}}$ also regularizes LLM's generation format as a list of features based on $\mathcal{T}$. s is further parsed as the feature list $F = [f_1, f_2, ..., f_{|F|}]$ representing recommended features. Then the ranking score of each item i is calculated as:

$$Score_i = |i^C \cap F| \tag{5}$$

Then items with Top-k highest ranking scores are retrieved from the item pool and recommended to users. In summary, we designed a framework TAXREC, which uses a taxonomy as the intermediate, to unify the representation of items throughout the recommendation pipeline. TAXREC can retrieve LLM's knowledge for zero-shot recommendations without any training and other users' interactions with item.

4 Experiments

This section empirically evaluates TAXREC by answering the following research questions (RQs):

- **RQ1**: How does TAXREC perform compared with current LLM-based zero-shot recommendation models?
- **RQ2**: How do the different components in TAXREC influence its effectiveness?
- **RQ3**: How do the key parameters affect the performance of TAXREC?

4.1 Experimental Setup

4.1.1 Datasets

We evaluate TAXREC on two widely used datasets for recommender systems:

- **Movie**: This is a movie recommendation dataset processed from MovieLens-100k[1] (Harper and Konstan, 2015), which is a widely utilized benchmark in the field of recommender systems. We follow (Bao et al., 2023) to set the 10 interactions before the target item as historical interactions. As we conduct experiments in a zero-shot setting which only infers LLMs, we don't need to split the dataset and randomly sample 2,000 instances from the original dataset for testing. For this dataset, the total number of items is 1,682.
- **Book**: This is a book recommendation dataset processed from BookCrossing[2] (Ziegler et al., 2005). The BookCrossing dataset contains some textual information about books, such as titles, authors, and publishers. Since this dataset lacks interaction timestamps, we can only construct historical interaction by random sampling. Therefore, we follow (Bao et al., 2023) to randomly select an item interacted by a user as the target item, and sample 10 items as the historical interactions. Similar to the movie dataset, we randomly sample 2,000 sequences for evaluation. The total number of items in this dataset is 4,389.

4.1.2 Baselines

To demonstrate the effectiveness of our model, we compare TAXREC against several state-of-the-art zero-shot recommenders:

RecFormer (Li et al., 2023): RecFormer encodes items as sentences and treats user histories as sequences of these sentences. We adopt the pre-trained model provided by the authors to make the recommendation as we aim at zero-shot scenarios.

UniSRec (Hou et al., 2022): UniSRec uses textual item representations from a pre-trained language model and adapts to a new domain using an MoE-enhance adaptor. Since we investigate the zero-shot scenario, we don't fine-tune the model and initialize the model with the pre-trained parameters provided by the authors.

ZESRec (Ding et al., 2022): It encodes item texts with a pre-trained language model as item features. Since we investigate the zero-shot scenario, for a fair comparison, we use the pre-trained BERT embeddings and do not fine-tune the model.

Popularity: This baseline recommends items based on their global popularity. It's a common baseline in recommender systems as it works well in cases where users prefer popular items. It's simple but can be strong in some domains.

AverageEmb: This baseline recommends the most similar items to a user based on the inner product between the user embedding and item embedding. The item embedding is obtained from pre-trained BERT, and the user embedding is the average of the user's historical items.

DirectLLMRec: This is a variant of our proposed TAXREC. In this method, we feed the user's his-

[1] https://grouplens.org/datasets/movielens/100k/

[2] https://github.com/ashwanidv100/Recommendation-System—Book-Crossing-Dataset/tree/master/BX-CSV-Dump

Table 2: Performance comparison between different zero-shot recommendation baselines and TAXREC. We report Recall(R) and NDCG(N) @(1, 5, 10) results multiplied by 10. The boldface indicates the best result and the underlined indicates the second best. All TAXREC results are significantly better than the baselines with $p < 0.05$.

Datasets	Methods	R@1	R@5	R@10	N@1	N@5	N@10
Movie	Popularity	0.005	0.035	0.160	0.005	0.020	0.061
	AvgEmb	0.000	0.040	0.100	0.000	0.020	0.039
	ZESRec	0.032	0.095	<u>0.222</u>	0.032	0.059	0.099
	UniSRec	0.032	0.063	0.143	0.032	0.048	0.074
	RecFormer	0.016	<u>0.141</u>	0.219	0.016	0.077	0.103
	DirectRec-LLaMA2	0.033	0.058	0.085	0.033	0.042	0.051
	TAXREC-LLaMA2	<u>0.045</u>	0.126	0.190	<u>0.045</u>	<u>0.095</u>	<u>0.148</u>
	DirectRec-GPT4	<u>0.045</u>	0.100	0.180	<u>0.045</u>	0.074	0.099
	TAXREC-GPT4	**0.060**	**0.175**	**0.300**	**0.060**	**0.117**	**0.157**
Book	Popularity	0.030	0.070	<u>0.155</u>	0.030	0.046	0.073
	AvgEmb	0.005	0.075	0.115	0.005	0.038	0.051
	ZESRec	0.005	0.070	0.115	0.005	0.037	0.051
	UniSRec	0.000	0.050	0.085	0.000	0.025	0.035
	RecFormer	0.010	0.060	0.125	0.010	0.033	0.054
	DirectRec-LLaMA2	0.001	0.010	0.015	0.001	0.004	0.006
	TAXREC-LLaMA2	<u>0.040</u>	<u>0.099</u>	0.150	<u>0.040</u>	<u>0.072</u>	<u>0.109</u>
	DirectRec-GPT4	0.000	0.015	0.025	0.000	0.006	0.010
	TAXREC-GPT4	**0.070**	**0.150**	**0.240**	**0.070**	**0.109**	**0.138**

torical items to LLM and ask LLM to generate the recommended items directly. This baseline tests the ability of LLM as a recommender without our proposed taxonomy framework.

4.1.3 Evaluation Metrics

Since TAXREC aims to generate the items that align with user preference, we adopt two popular evaluation metrics used in recommendation: Recall and Normalized Discounted Cumulative Gain (NDCG). We evaluate models' Top-K performance when k is selected as (1, 5, 10), separately.

4.1.4 Implementation Details

To ensure consistent sequence lengths, we pad historical interaction sequences shorter than the threshold (10) with the user's most recent interaction. For the LLM evaluation, we use both closed-source (GPT-4 via OpenAI's API) and open-source (LLaMA-2-7b with pre-trained parameters), which are widely adopted. Each experiment is repeated three times, and the average results are reported.

4.2 Overall Performance (RQ1)

In this section, we evaluate the recommendation performance of various methods in a zero-shot setting, which allows us to assess how LLMs can be utilized as recommenders without parameter tuning. The results are presented in Table 2. We compare our proposed TAXREC with two categories of models: traditional pre-trained zero-shot recommendation models (above the line) and LLM-based zero-shot models (below the line).

The following key observations can be drawn from the table: (1) Our proposed TAXREC significantly outperforms both traditional and LLM-based methods, particularly when applied to GPT-4, showcasing the effectiveness of prompting LLMs with our taxonomy framework in a zero-shot scenario. TAXREC leverages the LLM's internal knowledge to facilitate recommendation generation without relying on external information, thus unifying the recommendation task with the NLP task. (2) The LLM-based zero-shot method DirectRec shows limited recommendation capability. For instance, while DirectRec performs comparably to traditional models on the Movie dataset, it struggles on the Book dataset, where it barely produces correct recommendations. This suggests that LLMs perform better in domains they have encountered before, like Movies, but face challenges in unfamiliar domains, such as Books. However, by applying

Table 3: Performance of different component design variants of TAXREC.

Variant	Movie		Book	
	R@10	N@10	R@10	N@10
w/o Tax	0.112	0.078	0.025	0.010
w/o Match	0.254	0.127	0.165	0.100
TAXREC	**0.300**	**0.157**	**0.265**	**0.132**

our taxonomy framework, LLMs achieve substantially better performance—nearly ten times higher than DirectRec on the Book dataset. These findings highlight the gap between language tasks and recommendation tasks when using LLMs, reinforcing the importance of our study. Furthermore, it demonstrates how our taxonomy approach unlocks the potential of LLMs for recommendation tasks. (3) The performance improvements of TAXREC vary depending on the underlying LLM. For example, the improvement of TAXREC-GPT4 over DirectRec-GPT4 is more pronounced than the improvement of TAXREC-LLaMA2 over DirectRec-LLaMA2. This could be attributed to the inherent capabilities of different LLMs, such as comprehension and generation. Despite these differences, our proposed TAXREC consistently enhances the performance of direct recommendations by LLMs, underscoring its effectiveness regardless of the LLM used.

4.3 Ablation and Effectiveness Analysis (RQ2)

In this section, we conduct ablation studies on TAXREC to analyze the effectiveness of its ***component design*** and ***prompt design***.

Component Design. TAXREC consists of two key components: taxonomy regularization and feature-based matching. We perform ablation experiments by separately removing each component, with results shown in Table 3. In the "w/o Tax" variant, LLMs generate recommendations based solely on the user's original historical interactions, without the taxonomy. The "w/o Match" variant excludes the taxonomy-instructed matching mechanism and instead directly maps LLM-generated text to the original item pool.

The results show that the "w/o Tax" variant performs significantly worse than TAXREC. In the Movie dataset, "w/o Tax" achieves only half of TAXREC's performance, while in the Book dataset, where LLMs may have limited prior knowledge, performance drops by nearly tenfold compared to TAXREC. These results highlight the crucial role of taxonomy in LLM-based recommendation. The taxonomy helps retrieve the LLM's internal knowledge more effectively, enhancing its ability to perform recommendation tasks.

Table 4: Prompt design variants of TAXREC. "h w/ t" refers to history with title, "h w/o t" to history without title; "rec w/ title" refers to recommendation with title, and "rec w/o title" to recommendation without title. Results are Recall@10.

Variant	Movie		Book	
	h w/ t	h w/o t	h w/ t	h w/o t
rec w/ title	0.235	0.265	**0.240**	0.200
rec w/o title	**0.300**	0.180	0.225	0.025

Although taxonomy retrieval enhances LLM performance, the raw outputs from LLMs are still unstructured text. Table 3 shows that the absence of our parsing and matching mechanism ("w/o Match") results in reduced performance in both datasets. Without structured parsing, direct similarity calculations and mappings to candidate items cause LLMs to lose important information. By parsing outputs into the taxonomy format and matching them with the categorized item pool, recommendation accuracy is significantly improved. This demonstrates that taxonomy-instructed matching further boosts TAXREC's performance, underscoring its effectiveness.

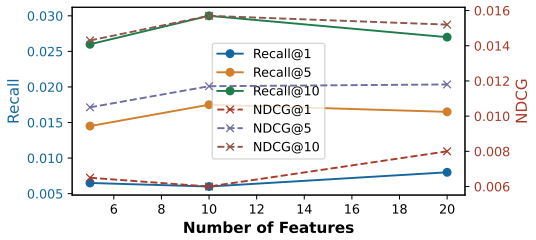
Prompt Design. We also experimented with different prompt templates to optimize TAXREC. Our framework uses three key prompts: the Taxonomy Generation Prompt, the Categorization Prompt, and the Recommendation Prompt. Table 4 summarizes various prompt configurations we tested, such as including or excluding item titles in the historical sequences and recommendations.

The results show that the optimal prompt design can vary across datasets. For the Movie dataset, the best combination was representing the history with titles but generating recommendations without titles. These variations emphasize the importance of prompt design in achieving optimal performance.

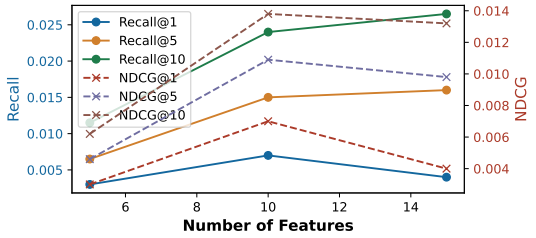
4.4 Hyperparameter Analysis (RQ3)

In TAXREC, two hyperparameters have a significant impact on recommendation performance: (1) the number of features in the taxonomy, and (2) the method for calculating the matching score.

Number of Features in Taxonomy. TAXREC improves LLM-based recommendations using an in-

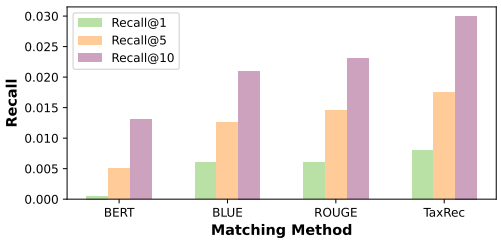


(a) Movie

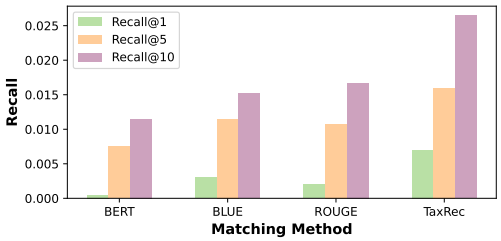


(b) Book

Figure 3: Recommendation performance by changing the number of features in taxonomy on both datasets.



(a) Movie



(b) Book

Figure 4: Recommendation performance by changing the methods for calculating the matching score on both Movie and Book datasets.

termediate taxonomy of features, where the number of features is key to performance. Since the taxonomy is generated by LLMs, the feature count may vary across datasets. Varying the number of features, we observe that: (1) Generally, more features lead to better performance, as they provide richer item representation and leverage more domain knowledge from LLMs. For instance, using just 5 features results in the lowest performance across metrics. (2) However, too many features can reduce performance. In Figure 3(a), Recall@5 and Recall@10 drop slightly when moving from 10 to 20 features. Similarly, in Figure 3(b), NDCG declines with 15 features compared to 10. This suggests that excessive features may introduce noise by exceeding the LLM's domain knowledge.

Methods for Matching Score. The matching component is essential in our TAXREC, in which the method that calculates the matching score is the key. We evaluate two types of methods: learning-based methods and rule-based methods. BERT embedding is a representative learning-based method, capturing semantic similarities using pre-trained models. In contrast, rule-based methods like BLUE, ROUGE, and our taxonomy-instructed mechanism use predefined rules to assess similarity.

Figure 4 presents the results: (1) The learning-based method, i.e., BERT embeddings, performs poorly on both datasets. This is likely because the model is not pre-trained on our specific dataset, and semantic similarity is not the focus of our task. While learning-based methods can excel when pre-trained or fine-tuned on a specific dataset, such training is resource-intensive and time-consuming. (2) In contrast, rule-based methods are better suited to TAXREC. Since TAXREC structures both the LLM outputs and the candidate item pool using a taxonomy, the representations are composed of fragmented rather than purely semantic information. (3) Among the rule-based methods, our proposed taxonomy-instructed matching mechanism performs best. This is because it aligns directly with the taxonomy, allowing LLM outputs to be easily parsed, and enabling similarity to be calculated through word-level matching without the need for complex rules.

5 Conclusions

In conclusion, our proposed method utilizing a taxonomy dictionary to enhance large language models (LLMs) for recommender systems demonstrates substantial improvements in recommendation quality and efficiency. By systematically categorizing and organizing items through a taxonomy framework, we address the key challenges faced by LLM-based recommendation systems, such as limited prompt length, unstructured item information, and uncontrolled generation. The incorporation of a taxonomy dictionary into the LLM prompts enables efficient token utilization and controlled feature generation, ensuring more accurate and con-

textually relevant recommendations. Experimental results show significant improvements over traditional zero-shot methods, demonstrating the efficacy of our approach and paving the way for further advancements in LLM-based recommendations.

6 Limitations

Despite the promising results of our taxonomy-based approach, several limitations should be acknowledged. First, there may be more effective methods to derive taxonomies beyond prompting LLMs, potentially capturing more detailed item nuances. Second, the LLMs' domain knowledge might be insufficient in some areas, affecting the quality of the taxonomy and recommendations. Lastly, the taxonomy generated via LLM prompts may lack completeness and scientific rigor, necessitating more scientifically grounded and systematically developed classification standards for greater accuracy and reliability.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-07-04, NSF awards (SaTC-2241068, IIS-2339198, III-2106758, and POSE-2346158), a Cisco Research Award, and a Microsoft Accelerate Foundation Models Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security and the National Science of Foundation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. *arXiv preprint arXiv:2305.00447*.

Yuwei Cao, Nikhil Mehta, Xinyang Yi, Raghunandan Keshavan, Lukasz Heldt, Lichan Hong, Ed H Chi, and Maheswaran Sathiamoorthy. 2024. Aligning large language models with recommendation knowledge. *arXiv preprint arXiv:2404.00245*.

Yuwei Cao, Liangwei Yang, Chen Wang, Zhiwei Liu, Hao Peng, Chenyu You, and Philip S Yu. 2023. Multi-task item-attribute graph pre-training for strict cold-start item recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 322–333.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1126–1132.

Hao Ding, Anoop Deoras, Yuyang (Bernie) Wang, and Hao Wang. 2022. Zero shot recommender systems. In *ICLR 2022 Workshop on Deep Generative Models for Highly Structured Data*.

Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, et al. 2022. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 13(1):3094.

Shanshan Feng, Haoming Lyu, Caishun Chen, and Yew-Soon Ong. 2024. Where to move next: Zero-shot generalization of llms for next poi recommendation. *arXiv preprint arXiv:2404.01855*.

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 720–730.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers

for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1754–1763.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Arka Pal, Deep Karkhanis, Manley Roberts, Samuel Dooley, Arvind Sundararajan, and Siddartha Naidu. 2023. Giraffe: Adventures in expanding context lengths in llms. *arXiv preprint arXiv:2308.10882*.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*.

Chen Wang, Ziwei Fan, Liangwei Yang, Mingdai Yang, Xiaolong Liu, Zhiwei Liu, and Philip Yu. 2024a. Pretraining with transferable attention for addressing market shifts in cross-market sequential recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2970–2979.

Chen Wang, Fangxin Wang, Ruocheng Guo, Yueqing Liang, Kay Liu, and Philip S Yu. 2024b. Confidence-aware fine-tuning of sequential recommendation systems via conformal prediction. *arXiv preprint arXiv:2402.08976*.

Chen Wang, Liangwei Yang, Zhiwei Liu, Xiaolong Liu, Mingdai Yang, Yueqing Liang, and Philip S Yu. 2024c. Collaborative alignment for recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2315–2325.

Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*.

Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Y Zhang, Qing Cui, et al. 2023a. Enhancing recommender systems with large language model reasoning graphs. *arXiv preprint arXiv:2308.10835*.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023b. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*.

Yu Wang, Zhiwei Liu, Jianguo Zhang, Weiran Yao, Shelby Heinecke, and Philip S Yu. 2023c. Drdt: Dynamic reflection with divergent thinking for llm-based sequential recommendation. *arXiv preprint arXiv:2312.11336*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*.

Shitao Xiao, Zheng Liu, Yingxia Shao, Tao Di, Bhuvan Middha, Fangzhao Wu, and Xing Xie. 2022. Training large-scale news recommenders with pretrained language models in the loop. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4215–4225.

Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2024. Unified pretraining for recommendation via task hypergraphs. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 891–900.

Shaowei Yao, Jiwei Tan, Xi Chen, Juhao Zhang, Xiaoyi Zeng, and Keping Yang. 2022. Reprbert: distilling bert to an efficient representation-based relevance model for e-commerce. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4363–4371.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. Lm-critic: Language models for unsupervised grammatical error correction. *arXiv preprint arXiv:2109.06822*.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2639–2649.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Song Zhang, Nan Zheng, and Danli Wang. 2022. Gbert: pre-training user representations for ephemeral group recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2631–2639.

Weizhi Zhang, Liangwei Yang, Yuwei Cao, Ke Xu, Yuanjie Zhu, and S Yu Philip. 2023b. Dual-teacher knowledge distillation for strict cold-start recommendation. In *2023 IEEE International Conference on Big Data (BigData)*, pages 483–492. IEEE.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32.