Taxonomy-Guided Zero-Shot Recommendations with LLMs

Anonymous ACL submission

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 method using a taxonomy dictionary. This method provides a systematic framework for categorizing and organizing items, improving the clarity and structure of item information. By incorporating the taxonomy dictionary into LLM prompts, we achieve efficient token utilization and controlled feature generation, leading to more accurate and contextually relevant recommendations. Our Taxonomy-guided Recommendation (TaxRec) approach features a two-step process: one-time taxonomy categorization and LLM-based recommendation, enabling zero-shot recommendations without the need for domain-specific fine-tuning. Experimental results demonstrate TaxRec significantly enhances recommendation quality compared to traditional zero-shot approaches, showcasing its efficacy as personal recommender with LLMs. Code is available at `https://anonymous.4open.science/r/TaxRec`.

1 Introduction

Due to the emergent ability (Wei et al., 2022), large language models (LLMs) have triggered the purse of artificial general intelligence (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., 2023) and grammar checking (Yasunaga et al., 2021). Besides, 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 generation, it is also compute intensive and mostly infeasible to calculate the ranking score for all candidate items within the pool.

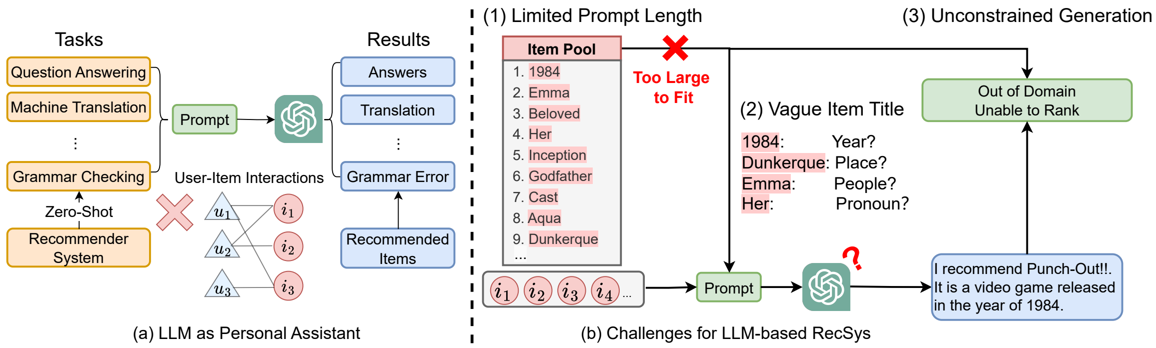


Figure 1: (a) LLMs are zero-shot recommenders without knowing other user-item interactions when acting as personal recommendation assistant. (b) Three challenges occur when integrating RecSys within the all-task-in-one LLM assistant, i.e., limited prompt length, vague item title and unconstrained generation.

In this paper, we propose using taxonomy dictionary to cope with previously mentioned challenges. Taxonomy is able to provide a systematic framework for identifying, organizing, and grouping items. For each dataset, we first retrieve the taxonomy dictionary from LLM to obtain the categorization knowledge of the domain and categorize all candidate items into structured item pool. It alleviates the vague item title/description problem by providing more item context information. For example, the item "1984" is more clarified as "Type: Book, Genre: Fiction, Theme: Power, ...". When prompting LLM for recommendation, we add the taxonomy dictionary within the prompt to inform LLM the candidate items information.

The taxonomy dictionary is a condensed categorization of 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 feed 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 taxonomy and categorized item pool. The second is LLM-based Recommendation step to inference user's preference based on the 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 TaxRecover current zero-shot recommenders, proving the effectiveness of our proposed item taxonomy categorization.

2 Related Work

2.1 LLM for Recommendations

Recommendation systems play a critical role in assisting users in finding relevant and personalized items or content. With the emergence of LLMs in recent years, there has been a growing interest in harnessing the power of these models to enhance recommendation systems. LLM-based recommender systems can be mainly categorized into two types: discriminative and generative (Wu et al., 2023). The core idea of discriminative LLM-based recommender systems is to utilize discriminative LLMs to learn better representations of users and items from contextual information for downstream tasks (Hou et al., 2022; Li et al., 2023; Xiao et al., 2022; Zhang et al., 2022; Yuan et al., 2023; Yao et al., 2022). Unlike discriminative methods, generative LLM-based recommendation systems

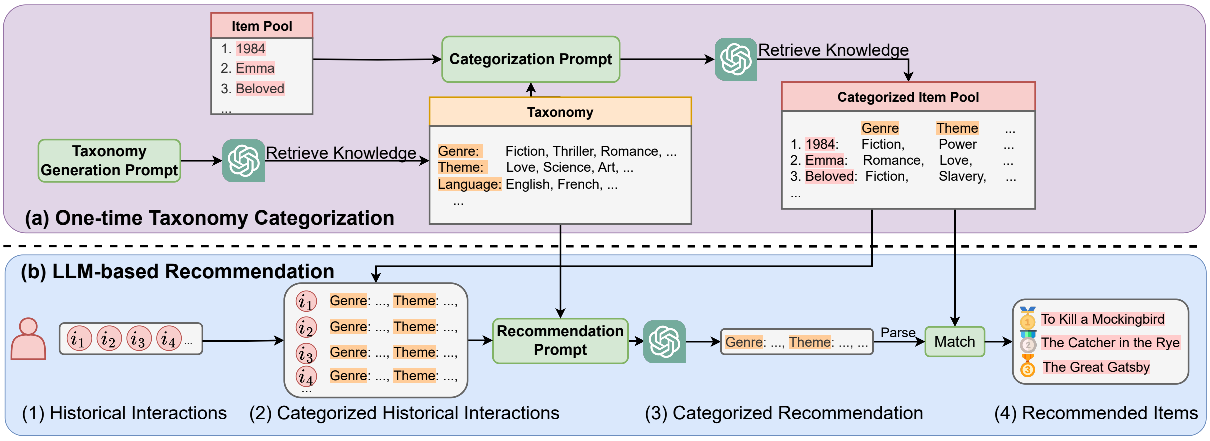


Figure 2: The proposed TaxRec for zero-shot LLM-based recommendation. (a) One-time Taxonomy Categorization step aims to generate in-domain taxonomy and enrich/categorize item's title into structured text information. (b) LLM-based Recommendation step provide ranked item lists for users based on the user's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

leverage the generative ability of LLMs to make recommendations. They achieve this by translating the traditional ranking-based recommendation tasks into natural language tasks. Compared with score computing and ranking strategies, they apply techniques such as prompt tuning and in-context learning to make LLMs able to generate the recommendations directly. (Geng et al., 2022) is one of the first text-to-text generative recommendation methods, which used the pre-trained T5 as the backbone. Recently, many methods explored the possibility of using LLMs to generate 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 database to LLMs to serve as autonomous agents for multiple recommendation tasks. (Wang et al., 2023a) used LLMs to augment the user-item interaction graph for recommendation. (Lyu et al., 2023) conducted a comprehensive investigation on how different types of prompts will influence the recommendation results. While (Wang et al., 2023c) proposed a multi-round self-reflection framework for LLM-based sequential recommendation. However, previous works did not consider the problem that text representations of items are vague and unstructured, which will hinder LLM's recommendation ability. In this paper, we propose a taxonomy-guided framework to solve this problem.

2.2 Zero-shot Recommendations

Zero-shot recommendation has emerged as a significant area in the field of recommendation systems. Unlike traditional systems that rely on model training, zero-shot recommendation systems aim to predict user preferences without parameter adjustment. Various methodologies have been explored to address this challenge. For instance, a content-based approach leverages item attributes (Lian et al., 2018). Other methods utilize pre-trained language models and embeddings to capture item and user characteristics from textual information (Ding et al., 2022; Hou et al., 2022; Li et al., 2023). With the rise of LLMs in recent years, several studies have explored their potential for making 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 faced the issue of limited context length in LLMs, which prevents the input of all items. Some approaches addressed this by using external tools or information (Wang et al., 2023b; Feng et al., 2024), while others employed a plug-in recommendation model to pre-select a small candidate pool (Hou et al., 2024; Wang and Lim, 2023; He et al., 2023). None of these methods have been able to solve this problem solely using the knowledge from LLMs. This paper proposes a taxonomy-guided LLM recommender that compresses the entire item pool using an LLM-generated taxonomy, enabling zero-shot recommendations without relying on 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 Tok-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 be then represented as designing a LLM-based function as:

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

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 $\mathcal{I^C} = \{i_j^C\}_{j=1}^{\mathcal{I^C}}$ and categorized historical interactions $\mathcal{H^C} = \{i_1^C, i_2^C, ..., i_{\mathcal{H^C}}^C\}$.

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:

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

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:

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

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 origional 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 $\mathcal{H^C}$>

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 $\mathcal{H}^C$ 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 $\mathcal{H}^C$ with taxonomy $\mathcal{T}$ to form a prompt to obtain the categorized recommendation as:

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

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 fits 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 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**[1]: This is a movie recommendation dataset processed from MovieLens-100k (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**[2]: This is a book recommendation dataset processed from BookCrossing (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

[1]https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/data

[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 and NDCG @(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 t-test $p < 0.05$.

Datasets	Methods	Recall@1	Recall@5	Recall@10	NDCG@1	NDCG@5	NDCG@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	0.222	0.032	0.059	0.099
	UniSRec	0.032	0.063	0.143	0.032	0.048	0.074
	RecFormer	0.016	0.141	0.219	0.016	0.077	0.103
	DirectRec	0.045	0.100	0.180	0.045	0.074	0.099
	TaxRec	**0.060**	**0.175**	**0.300**	**0.060**	**0.117**	**0.157**
Book	Popularity	0.030	0.070	0.155	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	0.000	0.015	0.025	0.000	0.006	0.010
	TaxRec	**0.070**	**0.150**	**0.240**	**0.070**	**0.109**	**0.138**

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 historical 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 uniform sequence lengths, we use the user's last interacted item to pad the historical interaction sequences with lengths < the threshold, 10. Because we are studying the ability to use LLM as a personal recommendation assistant, we focused on close-source and API-based LLMs in the evaluation to fit our scenario. Specifically, we evaluate the widely used GPT-4 by OpenAI's API, and conduct each experiment three times and present the average results.

4.2 Overall Performance (RQ1)

In this section, we aim to investigate the recommendation performance of various methods under the zero-shot setting, which enables us to evaluate how LLMs can be used as recommenders without tuning parameters or any historical user-item interactions. The evaluation results are presented in Table 2. We compare our proposed TaxRec with two types of models: traditional pre-trained zero-shot recommendation models and LLM-based zero-shot models, as above the line and under the line respectively.

From the table, we can draw the following observations: (1) Our proposed TaxRec significantly outperforms both traditional and LLM-based methods, demonstrating the superiority of prompting LLM with our proposed taxonomy framework to make recommendations in the zero-shot scenario. TaxRec successfully retrieves LLM's knowledge to facilitate the generation ability for recommendation task without any knowledge outside of LLMs. In this way, TaxRec successfully unifies the recommendation task to the NLP task. (2) LLM-based zero-shot method, i.e., DirectRec, has limited recommendation ability. For example, in the Movie dataset, DirectRec achieves comparable performance compared with traditional methods, however, in the Book dataset, it can hardly make correct recommendations. We can infer that in some domains that LLMs have seen before, such as Movie, their performance can be similar to traditional pre-trained models. While in some domains that LLMs have not seen before, such as Book, their recommendation capability is impeded. Nevertheless,

Table 3: Ablation Study of TaxRec

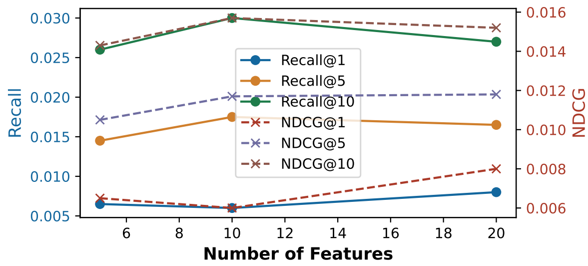
Variant	Movie		Book	
	R@10	N@10	R@10	N@10
w/o Tax	0.112	0.078	0.025	0.010
w/o Match	0.254	0.127	0.165	0.100
TaxRec	**0.300**	**0.157**	**0.265**	**0.132**

after applying our proposed taxonomy method, LLMs can achieve significantly better performance, almost 10 times greater than DirectRec in Book dataset. These results show that there is still a gap between language task and recommendation task when using LLMs, which indicates the importance of our study. Additionally, it demonstrates that our taxonomy approach unlocks the potential of LLMs on recommendation tasks. (3) Some traditional pre-trained recommendation models can achieve a fair performance. For example, RecFormer and ZESRec perform well in Movie dataset, and Popularity performs the second best in Book dataset. This implies that each domain may have different characteristics and thus be suitable for different methods. However, our proposed TaxRec performs best for both datasets, showing that it has fully tapped into LLM's profound knowledge.

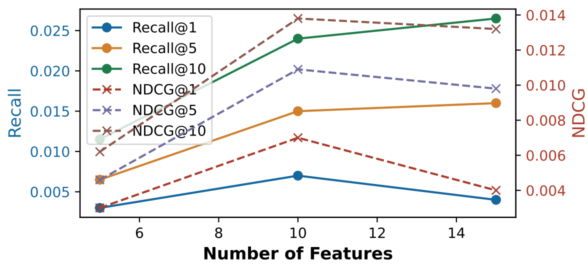
4.3 Ablation and Effectiveness Analysis (RQ2)

This section conducts ablation experiments on TaxRecby removing taxonomy regularization and the feature-based matching separately. The results are shown in Table 3. "w/o Tax" uses LLMs to make recommendations given user's original historical interactions without taxonomy. "w/o Match" is TaxRec without taxonomy-instructed matching mechanism, which directly maps LLM's generated text to the original item pool.

To investigate how taxonomy can help retrieve LLM's knowledge, we conduct an experiment on TaxRec's variant "w/o Tax", which uses LLMs to make recommendations without taxonomy (only with user's original historical interactions). From this table, we can observe that the performance of "w/o Tax" is significantly lower than TaxRec. In the Movie dataset, "w/o Tax" can just achieve half of the ability of TaxRec, while in the Book dataset which LLMs may not have good knowledge of, we can see a 10-times drop in the performance of "w/o Tax" compared with TaxRec. These results show that our designed taxonomy method plays an essential role in LLM recommendation. With



(a) Movie



(b) Book

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

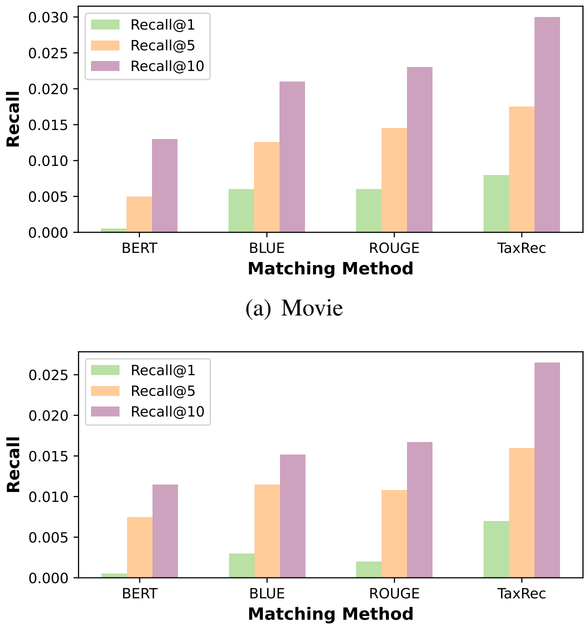
the help of taxonomy, LLM-embedded knowledge can be better retrieved, which makes great use of LLM's capability for recommendation tasks.

Though taxonomy can help us retrieve LLM's knowledge, the outputs of LLMs are still plain text. From Table 3, we can find that without our parsing and matching mechanism, i.e., "w/o Match", the recommendation performance will decrease for both datasets. Because the generated text is unstructured strings, directly calculating the similarity and mapping it to the candidate items will make LLMs confused and lose some information. However, after parsing the outputs to the taxonomy's format and matching them with the categorized item pool in the same format, the recommendation performance is better. These results demonstrate that when paired with the taxonomy-instructed matching, TaxRec will have better performance, implying the effectiveness of the matching.

4.4 Hyperarameter Analysis (RQ3)

In our proposed TaxRec, there are two hyperparameters that make important impacts on recommendation performance: 1) the number of features in the taxonomy, and 2) the methods for calculating the matching score.

Number of Features in Taxonomy. Since our proposed TaxRec improves LLM recommendation by using an intermediate taxonomy, where the taxonomy is a list of features, the number of features is



(a) Movie

(b) Book

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

key to performance. As the taxonomy is generated by LLMs, the total number of features may differ per domain (dataset). To investigate this hyperparameter's impact, we vary the number of features for each dataset and present the results in Figure 3. We find that: (1) Generally, more features lead to better recommendation performance. For example, TaxRec with a 5-feature taxonomy yields the lowest results across both metrics, as more features enrich item representation and retrieve more domain knowledge from LLMs, enhancing recommendations. (2) However, more features do not always equate to better performance. In Figure 3(a), Recall@5 and Recall@10 slightly decrease when using 20 features compared to 10. Similarly, in Figure 3(b), NDCGs drop when using 15 features instead of 10. We infer that exceeding the LLMs' domain knowledge leads to some features being assigned random values, introducing noise and reducing recommendation quality compared to using fewer, more relevant features.

Methods for Matching Score. The matching component is essential in our TaxRec, in which the method that calculates the matching score is the key. To investigate the effect of it, we examine two different types of methods, the learning-based method and the rule-based method. BERT embedding is a representative learning-based method that can be used to calculate the matching score. It leverages the contextual understanding of pre-trained models and can capture semantic similarities between texts. While BLUE score, ROUGE score, and our proposed taxonomy-instructed matching mechanism are rule-based methods. They rely on predefined rules and algorithms to evaluate the similarity, each with a different focus. Figure 4 shows the results of using these methods. From the figure, we can have the following findings: (1) The learning-based method, i.e., BERT embedding, does not perform well on both datasets. This is because the model is not pre-trained on our dataset, and the semantic information is not our focus. The learning-based method can perform well if they are well pre-trained or fine-tuned on a specific dataset, however, the training will cost some time and resources. (2) Instead, rule-based methods are more suitable for TaxRec. The reason is that TaxRec uses a taxonomy to structure both LLM's outputs and the candidate item pool, thus the representations for them are mixed with fragmented information rather than semantic information. (3) Among the rule-based methods, our proposed matching mechanism performs best. This is because this mechanism is instructed by the taxonomy which specifically aligns with our task. While instructed by taxonomy, the outputs generated by LLMs can be easily parsed and the similarity can be easily calculated by word-wised text matching without other sophisticated 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 contextually 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.

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*.

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*.

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.

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. 2023. 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.

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.