Tired of Plugins?
Large Language Models can be End-to-End Recommenders

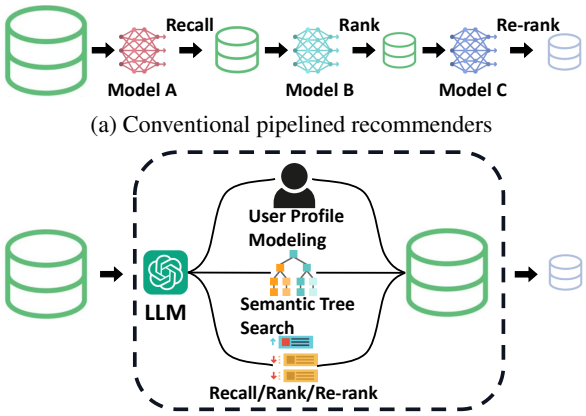
Anonymous ACL submission

Abstract

Recommender systems aim to predict user interest based on historical behavioral data. They are mainly designed in sequential pipelines, requiring lots of data to train different subsystems, and are hard to scale to new domains. Recently, Large Language Models (LLMs) have demonstrated remarkable generalized capabilities, enabling a singular model to tackle diverse recommendation tasks across various scenarios. Nonetheless, existing LLM-based recommendation systems utilize LLM purely for a *single task* of the recommendation pipeline. Besides, these systems face challenges in presenting *large-scale* item sets to LLMs in natural language format, due to the constraint of input length. To address these challenges, we introduce an LLM-based end-to-end recommendation framework: UniLLMRec. Specifically, UniLLMRec integrates multi-stage tasks (*e.g.,* recall, ranking, re-ranking) via *chain-of-recommendations*. To deal with large-scale items, we propose a novel strategy to structure all items into a *semantic item tree*, which can be dynamically updated and effectively retrieved. UniLLMRec shows promising zero-shot results compared to supervised models, and it is highly efficient by reducing 86% input tokens than LLM-based models. Our code is available to ease reproduction.[1]

1 Introduction

Recommender systems aim to understand the preferences, historical choices, and characteristics of users and items via collected behavioral data (*e.g.,* clicks, likes, pages viewed, and etc) (Bobadilla et al., 2013). Given their capability to predict user interests, recommender systems are widely adopted by content/product providers. The recommendation process includes item candidates retrieval (*i.e.,* recall) (Bobadilla et al., 2013), prioritization of potential items (*i.e.,* ranking) (Wang et al., 2020; Qi et al., 2021), and the curation of a diverse set of items for users (*i.e.,* re-ranking) (Pei et al., 2019). Conventional recommender systems often structure the recommendation process in sequential pipelines, as illustrated in Figure 1a. These systems comprise several specialized models, each tailored for one stage (*e.g.,* recall, ranking, re-ranking). Yet, the training and ongoing maintenance of several distinct models incur significant costs. Additionally, re-training the entire system with new data poses challenges to scalability and operational efficiency. Therefore, it is important to design a unified end-to-end model to alleviate these concerns. Meanwhile, understanding human behavior presents substantial challenges in conventional recommender systems. Typically, these systems (Kang and McAuley, 2018; Sun et al., 2019; Song et al., 2019) model user and item information in vector space, resulting in the potential loss of rich contextual semantics.



(a) Conventional pipelined recommenders

(b) Our proposed end-to-end recommender

Figure 1: Examples of conventional pipelined and LLM-empowered end-to-end recommender systems.

Recent emergence of Large Language Models (LLMs), such as ChatGPT (Brown et al., 2020) and Claude (Bai et al., 2022), has demonstrated robust ability to excel in a wide array of NLP tasks. The inherent potential of LLMs positions them as natural

[1] https://anonymous.4open.science/r/UniLLMRec-E7AB/

zero-shot solvers, capable of addressing multiple recommendation challenges simultaneously. Beyond task performance, LLMs exhibit impressive capacity to assimilate human knowledge related to our physical world and society, as highlighted in recent studies (Sanh et al., 2021; Wei et al., 2021; Wang et al., 2022). In light of these capabilities, LLMs have recently been applied to *enhance recommender systems with strong zero-shot abilities and deeper understanding of human behavior*.

Firstly, Dai et al. (2023); Petrov and Macdonald (2023) formulate recommendation tasks within the framework of natural language generation. In their approaches, LLMs are finetuned to cater to different recommendation scenarios via Parameter Efficient Fine Tuning (PEFT) methods such as LoRA and P-tuning (Hu et al., 2021; Liu et al., 2021). **Challenge 1 arises**: though claiming to be efficient, these fine-tuning techniques rely on substantial training data which is costly to obtain. This issue worsens in dynamic environments, where continual item updates drive periodic LLM training.

Meanwhile, Hou et al. (2023); Gao et al. (2023) design well-crafted prompts for the ranking stage of recommendation. These methods leverage on the strong zero-shot ability of LLM to perform ranking task. However, **challenge 2 arises**, as these methods under-utilize the strong general or multi-task capabilities of LLM, whereas we believe LLM can do much more beyond a single-stage application.

Moreover, Gao et al. (2023); Dai et al. (2023) attempt to concatenate the entire list of items into a single prompt to leverage LLM for recall. Items are often represented using attributes such as title, description, listing date, category, and etc. However, the efficacy of such systems is constrained by trade-off between the size of the item list and the amount of detail provided for each item. For instance, incorporating more comprehensive descriptions requires reducing the number of items included in the list, and vice versa. Consequently, existing systems limit their input to either an item title or a brief description, with the item size ranging typically from 10 to 100. **Challenge 3 arises**, reflecting the difficulty in presenting a large-scale item corpus to LLMs in the natural language format, due to the constraint of LLM input length.

To address the aforementioned challenges, we propose UniLLMRec, which utilizes one single LLM to execute items recall, ranking, and re-ranking in a unified end-to-end recommendation framework (see Figure 1b). Note that we leverage the zero-shot capability intrinsic to LLMs, thereby neither training nor finetuning is needed. Hence, UniLLMRec not only streamlines the recommendation process but also significantly mitigates the reliance on extensive training datasets, enabling more efficient and scalable deployment in various recommendation scenarios. To make UniLLMRec applicable to large-scale item corpus, we design a novel semantic tree recall strategy. Specifically, we construct a semantic item tree according to item semantic information (*e.g.,* category, subcategory, keywords, etc) among large-scale item lists. Note that each leaf node consists of only a small portion of entire items. With the constructed item tree, the recall of UniLLMRec is achieved by (i) traversing from the root node until it reaches the leaf node, and (ii) searching items from selected leaf nodes. In comparison, conventional methods search candidates from the entire item list. In summary, our contributions are in three-fold:

- We propose UniLLMRec, a first end-to-end LLM-empowered recommendation framework, which realizes the whole recommendation process including stages of recall, ranking, and re-ranking. It is much easier to deploy compared with conventional recommender systems.
- We design a hierarchical semantic tree structure that can frame large-scale item lists into small lists in leaf nodes. Such the semantic tree can be dynamically updated and effectively retrieved.
- We validate the effectiveness and efficiency of UniLLMRec on two benchmark datasets. Our zero-shot results are comparable to conventional baselines trained on voluminous data. Compared to LLM-based baselines, UniLLMRec is highly efficient by reducing 85% input tokens.

2 Proposed Framework

2.1 Overview

The overall framework of UniLLMRec is depicted in Figure 2. Firstly, we explain our strategy of constructing semantic item tree in Section 2.2. Based on our semantic item tree, UniLLMRec can traverse a large-scale item corpus for fast recall from leaf nodes. Next, we explain our end-to-end architecture of UniLLMRec in Section 2.3. We elaborate on how it can capture user interest and conduct recommendation stages from recall to re-ranking.

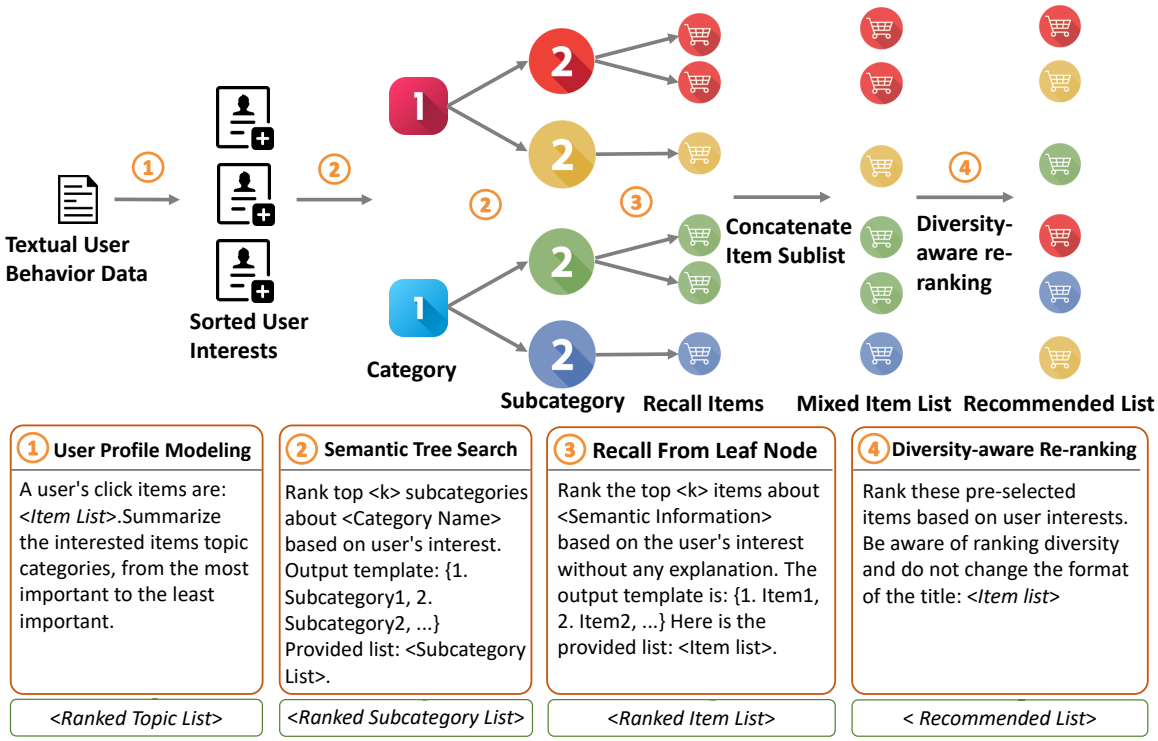


Figure 2: The overview of UniLLMRec: Unified LLM-empowered end-to-end recommendator

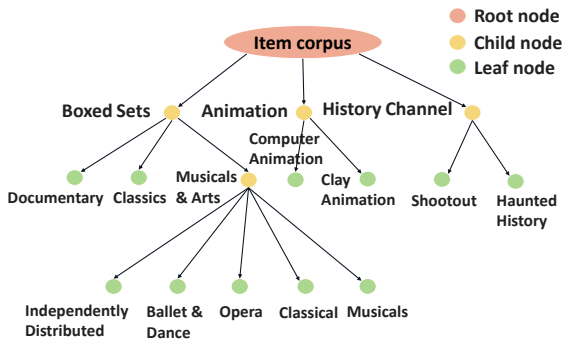


Figure 3: An example of Semantic Item Tree.

2.2 Semantic Item Tree Construction

LLM-based recommender systems face challenges in recalling items from large-scale item corpus mainly due to two challenges: (i) LLMs are constrained by limited input length, while the concatenated descriptions of 10-100 items can easily exceed such limit. (ii) lengthy item descriptions can easily confuse LLM, making it hard for LLM to extract salient item features. To alleviate these challenges, we introduce a hierarchical tree to organize items into leaf nodes, facilitating LLM in efficiently handling large-scale item sets. Figure 3 shows an example of a semantic item tree.

Specifically, the root node (depicted in red color) encompasses all the items within the item corpus. Starting from the root node, items are categorized into corresponding subsets according to their semantic information (*e.g.,* categories, subcategories, keywords, and other useful information if necessary). Each subset corresponds to a child node (depicted in yellow color) of its root node. For each node, we keep splitting it further into child nodes, if it contains more fine-grained sub-categories. The stopping criteria are met when (i) the attributes of this node are semantically self-contained and (ii) the number of items belonging to the node is reasonable. Such nodes (depicted in green color) are defined as leaf nodes, and each of them is a small subset of the large-scale item list.

2.3 LLM-empowered End-to-End Recommendation Framework

Existing LLM-based systems (Gao et al., 2023; Hou et al., 2023; Wang and Lim, 2023) mainly focus on the ranking stage in the recommender system, and they rank only a small number of candidate items. In comparison, UniLLMRec is a comprehensive framework that unitizes LLM to integrate multi-stage tasks (*e.g.,* recall, ranking, re-ranking) by chain-of-recommendation, referring to Section 2.3.1. Moreover, we elaborate on our effective retrieval strategy that enables UniLLMRec to recall related items among large-scale item sets in Section 2.3.2

2.3.1 Chain-of-Recommendation Strategy

With the aid of semantic item tree, we design a chain-of-recommendation strategy to seamlessly integrate it with our recommendation process. UniLLMRec provides an effective way for LLM to handle large-scale item sets under zero-shot setting. UniLLMRec executes the recommendation chain in a single session as follows:

User Profile Modeling. Since private user profile information (*e.g.,* age, gender, interests) is absent from the public dataset, we use user's interaction history as LLM input for user profile modeling.

Semantic Tree Search. UniLLMRec traverses the semantic tree from the root node to its child nodes. The search stops when the leaf node is reached. In each step, it deduces and ranks the top categories based on user interaction history and interest. More details are discussed in Section 2.3.2.

Recall from Leaf Node. Every leaf node corresponds to a small subset of items that cannot be further divided based on semantic information. Hence, the text describing all items in the subset can be easily fed into UniLLMRec. Then, UniLLMRec will recall top items by considering user interaction history and interest.

Diversity-aware Re-ranking. After recalling items from various leaf nodes, we aim to enhance the diversity of the recommendation items. Therefore, UniLLMRec will re-rank all the recalled items with our well-curated prompt to ensure diversity.

2.3.2 Search Strategy

The purpose of our search strategy is to balance between the diversity and relevance of retrieved items. In general, we apply Depth-first Search (DFS) on our semantic item tree, as demonstrated in Algorithm 1. In particular, throughout each step of the search, only the top-ranked nodes will be selected for further DFS search, allowing UniLLMRec to bypass less relevant nodes. Upon reaching a leaf node, UniLLMRec will recall top k items from the item subset of this leaf node. The search ends if either (i) all leaf nodes are traversed, or (ii) the desired number of n items has been recalled. The parameter k effectively serves a lever to modulate the diversity of recalled items. Opting for a smaller k increases the recommendation diversity, but at the cost of increased search time. Conversely, a larger k tends to reduce diversity while expediting the search process.

Algorithm 1: UniLLMRec

Input: User-item interaction history H
Output: Recommended item list R
Initialize: $L = \emptyset, Q =$ Queue()

```
Infer interests:
  I = User_profile_modeling(H)
  Q.push(root)
while |L| < n do
    node = Q.front()
    Q.pop()
    if node is leaf node then
        Get item subset from node:
          items =
          Recall_from_subset(H, I, subset, k)
        L.add(items)
    else
        childnodes =
          Semantic_tree_search(H, I, node)
        for node in childnodes do
            Q.push(node)
        end
    end
end
R = Diversity-aware Re-ranking(H, I, L)
```

3 Experiment

In this section, we will first introduce the experiment setting, then evaluate the model performance on recall and re-rank tasks, and finally conduct some topic analysis.

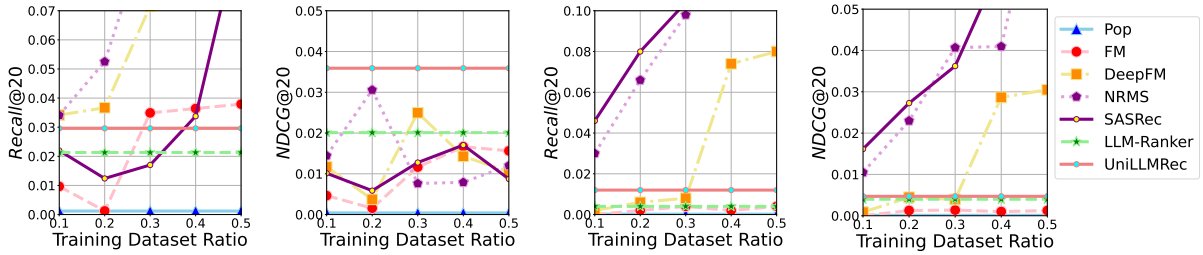
3.1 Experiment Setting

3.1.1 Datasets

In the experiments, we utilized two benchmark datasets including the MIND dataset (Wu et al., 2020) and Amazon Review dataset (He and McAuley, 2016) in the category of Movies and TV. To ensure fair comparisons in our experiments, all methods exclusively utilized the item titles as features. For the MIND dataset, 500 test instances are randomly sampled from the small_dev subset, while the small_train subset (51283 samples) served as the training set. For Amazon dataset, 500 samples were chosen from the reviews dataset for testing, with the remaining 70728 samples used for training. We list the statistics of datasets in Table 1.

3.1.2 Evaluation Metrics

We focus on evaluating the performance of the proposed framework and baseline in recall and



(a) Recall on MIND (b) NDCG on MIND (c) Recall on Amazon (d) NDCG on Amazon

Figure 4: Performance Comparison of Recall and NDCG value on MIND and Amazon datasets.

Dataset	Training set size	Test set size
MIND	51,283	500
Amazon	70,728	500

Table 1: The statistic detail of dataset

re-ranking tasks. For each model, we primarily consider its Recall metric and the Normalized Discounted Cumulative Gain (NDCG) in the recall task, and Intra-List Average Distance (ILAD) (Zhang and Hurley, 2008) in re-ranking tasks. We evaluate the above metrics in top-k recommendation items.

3.1.3 Baselines

UniLLMRec are compared with Popularity-based recommendation, FM (Rendle, 2010), DeepFM (Guo et al., 2017), NRMS (Wu et al., 2019), SASRec (Kang and McAuley, 2018), and LLM-Ranker (Hou et al., 2023).

3.1.4 Implementation Details

The UniLLMRec framework leverages **gpt-3.5-turbo**[2] as the backbone model of LLM. The semantic tree depth in MIND dataset is 2, with leaf nodes merely located in the second layer. There are 17 and 276 nodes in the first and second layers respectively. As for the Amazon dataset, items without semantic information are discarded during semantic tree construction. Subsequently, the constructed tree has a depth of 4, and the leaf nodes may be located in all layers. The node numbers from the first layer to the fourth layer are 78, 298, 126, and 19, respectively. In the semantic tree search stage, we set the recall subnode number as 10. Meanwhile, in the experiments, the parameter k in the recall stage serves to limit the number of selected leaf nodes and is set to 5. In addition, the total number of recalled items is set at 20.

[2] https://platform.openai.com/docs/models/gpt-3-5

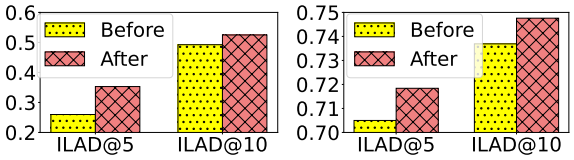
For conventional models, FM uses TF-IDF (Term Frequency-Inverse Document Frequency) (Salton and Buckley, 1988) of item titles as features, while in DeepFM, NRMS, and SASRec, item word embeddings are employed as features. More details on parameter setting can be found in **Appendix A.2**. Then, we increase the 10% training set size for each model until the model performance is equivalent to UniLLMRec. Thus, we can evaluate the performance between the capabilities of zero-shot end-to-end methods and supervised conventional recommendation models.

3.2 Performance Comparison

The overall performance of UniLLMRec and baselines are shown in Figure 4. To be specific, the proposed end-to-end LLMRec framework is compared with the methods in two categories.

The first is the method implemented under a zero-shot setting. The popularity-based method, hampered by the absence of user-specific information, demonstrated an exceedingly low recall of items. LLM-Ranker outperforms popularity-based methods in both Recall and NDCG metrics, yet it lags behind UniLLMRec. UniLLMRec is capable of refining the candidate set based on user interests and semantic trees, resulting in a smaller candidate set compared to LLM-Ranker, thereby leading to improved performance.

The other is the conventional recommendation model. Our main focus lies in evaluating how the performance of UniLLMRec is competitive with conventional recommendation models with varying amounts of training data. The performance comparison is shown in Figure 4 where x axis denotes the training dataset ratio and y axis denotes the $Recall@20$ and $NDCG@20$. While the training dataset ratio increases, FM excels UniLLMRec when r reaches 0.3 in $recall@20$ on MIND but fails to outperform UniLLMRec on Amazon. DeepFM, NRMS, and SASRec



(a) MIND (b) Amazon

Figure 5: The impact of the re-rank stage on the recommendation diversity.

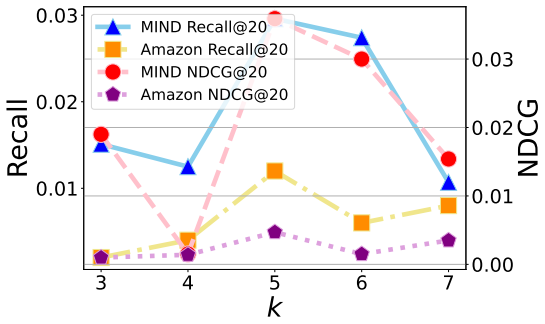


Figure 6: The impact of k on recall performance.

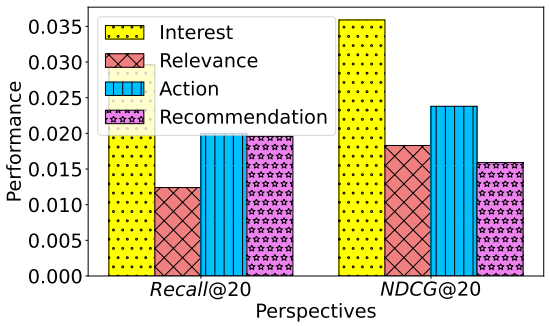


Figure 7: The impact of various perspective prompt design on recall performance.

outperform UniLLMRec in $Recall@20$ on both MIND and Amazon under a small ratio of training datasets. UniLLMRec outperforms baselines in $NDCG@20$ on MIND, which means it can recall the most relevant items in higher rank.

In addition, UniLLMRec achieves better performance on MIND than Amazon. Due to the imbalanced structure of the semantic tree in the Amazon dataset, as well as the issue of homogenized item titles and insufficient semantic information of items, UniLLMRec can not fully leverage its ability to recall items based on the semantic tree.

We also evaluate the impact of re-rank on diversity, and demonstrate the results in Figure 5, which indicates that the re-ranking step contributes to enhancing the diversity of recommended results.

3.3 Hyper-parameter Analysis

The recall number k in leaf nodes is the only hyper-parameter in UniLLMRec. We conducted a study on the impact of k on the recall task, and illustrate the results in Figure 6. As the value of k increases, the number of items recalled by our model from different leaf nodes steadily rises. We observe a phenomenon where both recall rate and NDCG initially rise and then decline with the increasing k. Clearly, with the continuous increment of k, the number of items recalled from each node also increases, indicating that the model tends to recommend items from subsets that are of higher interest to the user. When k decreases, the model recalls items from more leaf nodes, resulting in higher diversity in the retrieved results. In summary, the parameter k plays a crucial role in the model by influencing the trade-off between diversity and the quantity of recalled items under different categories.

3.4 Prompt Study

UniLLMRec utilizes prompts to adapt the LLM to recommendation tasks, where the design of prompt templates plays a crucial role in fully leveraging the capabilities of LLM in recommendation tasks. Therefore, we craft prompt templates from four different perspectives including interest, relevance, action, and recommendation tailored to the news recommendation as in Prompt4NR (Zhang and Wang, 2023). The prompts designed from various perspectives are detailed in Table 2 where the blue variant prompts are changed based on perspective. The performance of models under these four types of prompt settings is shown in Figure 7 from which we can see that prompt design significantly impacts the model performance. Using a relevance-based prompt yielded a recall rate of only 1.24% and an NDCG of 0.0183. By contrast, models using prompts of action and recommendation achieve approximately 2% recall rate. Besides, the best performance is observed under the interest-based prompt design, where the recall rate and NDCG were twice that of the relevance prompt model.

These results underscore the significance of prompt design on non-fine-tuned LLMs in recommendation tasks. The interest-based prompt design can effectively leverage the LLM's ability to uncover user interests, thereby enhancing the personalization and precision of recommendations.

3.5 Token Requirement Analysis

UniLLMRec recalls items from subsets based on the semantic tree, which effectively reduces the model's token requirement in the recall stage. We conduct a statistical analysis on the size of the can-

Table 2: Prompt design from 4 perspectives.

Perspective	User profile modeling	Semantic tree search	Recall from leaf node	Diversity-aware re-rank
Prompt template	A user's click items are: <Item List>. <Perspective-Variable Prompt>, from the most important to the least important.	Rank the top <k>subcategories about <Category Name>based on <Perspective-Variable Prompt> from the following candidates without any explanation. The output template is: {1. Subcategory1, 2. Subcategory2, ...} Here is the provided list: <Subcategory List>.	Rank the top <k>items about <Semantic Information>based on <Perspective-Variable Prompt> from the candidates about <Topic>without any explanation. The output template is: {1. Item1, 2. Item2, ...} Here is the provided list: <Item list>.	Rank these pre-selected items based on <Perspective-Variable Prompt>. Be aware of ranking diversity and do not change the format of the title: <Item list>.
interest	Summarize the interested items topic categories	the user's interest	the user's interest	the user's interest
relevance	Summarize the news topic categories related to users	the relevance related to the user	the relevance related to the user	the relevance related to the user
action	Summarize the news topic that the user are likely to click on	the probability that the user is likely to click	the probability that the user is likely to click	the probability that the user is likely to click
recommendation	Summarize the news topic worth recommending to the user	the degree of recommendation to the user	the degree of recommendation to the user	the degree of recommendation to the user

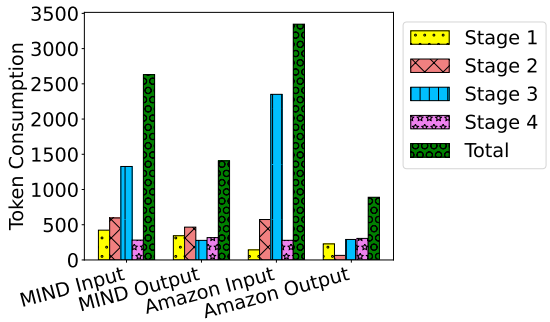


Figure 8: Consumption of tokens for each stage.

didate item set and the average token length for each item. After sampling, the MIND dataset comprises 1,217 items, while the Amazon dataset has 6,167 items, with an average token length of 14 and 10, respectively. The total tokens needed to input all items into the LLM exceed ten thousand. Consequently, an LLM recalling items from candidate sets faces inherent challenges in handling such large-scale item datasets.

However, by employing the UniLLMRec with semantic tree-based search, the token requirement can be effectively reduced. Figure 8 illustrates average token consumption in each framework stage, demonstrating fewer tokens used by UniLLMRec in the four stages. For a set of items with a scale of n, our model reduces the token requirement from $O(n)$ to $O(log(n))$, enabling LLM to process large-scale item sets.

3.6 Topic Analysis

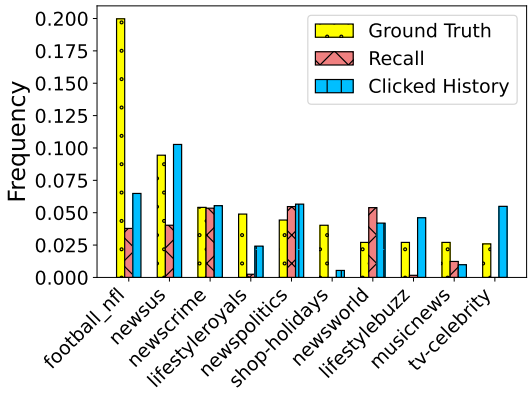
We conduct a statistical analysis on the distribution of topics from UniLLMRec, user interaction history, and ground truth. Subsequently, we calculate the similarity in topic distribution between ground truth and user click history, as well as ground truth and UniLLMRec. These values are 0.8091 and 0.6239, respectively, in the MIND dataset, and 0.8524 and 0.6157 on Amazon. Figure 9 illustrates the top 10 most frequently occurring topics in the ground truth. For the top 5 most frequent topics in the ground truth, UniLLMRec also demonstrates a high frequency in generating these topics. Additionally, UniLLMRec generates content from a broader range of topics, thereby enhancing the diversity of results.

4 Related Work

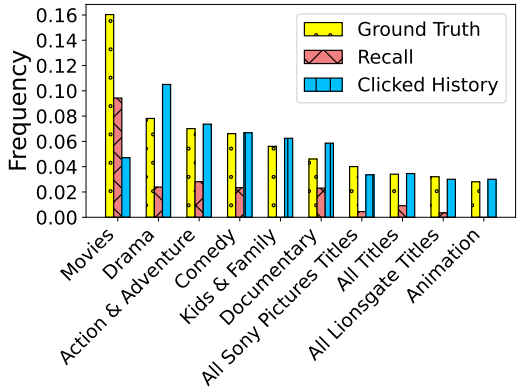
In this section, we provide an overview of the related work on conventional recommendation models and large language model for recommendation.

4.1 Conventional Recommendation Models

Conventional recommendation model (CRM) (Lin et al., 2023) usually consists of the following pro-



(a) MIND



(b) Amazon

Figure 9: Illustration on the topic distribution.

cedures. It takes historical user behaviors, user profile feature, and item feature as input. Then, the embedding layer maps the sparse feature into dense vectors. Next the feature interaction component is explicitly designed. For example, DeepFM (Guo et al., 2017) adopts the linear model, factorization machine, and fully connected network to capture the low and high order interactions among different feature fields. Finally, the output layer generates the prediction result of the recommendation task.

However, these methods are explicitly proposed for one specific stage of recommendation, thus can not be simultaneously deployed in different stages (from recall to re-ranking) in an end-to-end manner. By contrast, in this paper, we are the first to propose an LLM-enpowered end-to-end framework which leads to easy deployability with high efficiency.

4.2 Large Language Model for Recommendation

In recent years, large language models (LLMs) have shown their great potential and strong capability in handling different tasks like computer vision (Yang et al., 2023). In the recommendation community, existing methods incorporating LLMs can be categorized into two groups. On the one hand, some works directly generate the recommendation result of item ID (Geng et al., 2022; Cui et al., 2022; Hua et al., 2023). For example, P5 (Geng et al., 2022) reformulates recommendation tasks to natural language processing tasks utilizing personalized prompts and conducts conditional text generation. Hua *et al.* examine various item IDs based on P5 (Hua et al., 2023). Nonetheless, they require fine-tuning the LLMs which results in high computation costs even if some parameter-efficient fine-tuning techniques are adopted (Hu et al., 2021; Liu et al., 2021; Bao et al., 2023). On the other hand, some approaches (Gao et al., 2023; Liu et al., 2023; Hou et al., 2023) adopt a straightforward strategy of inputting the candidate set directly into the LLM to generate the recommended item list using hard prompts, without fine-tuning the LLM.

However, all the existing methods primarily focused on one specific recommendation stage, and they also face challenges related to input token limitations and susceptibility to noise information. By contrast, by introducing the semantic tree structure and search strategy, our proposed framework can tackle these problems and it also supports large-scale dynamically updated item corpus free from fine-tuning the LLM.

5 Conclusion

We propose UniLLMRec, the first end-to-end LLM-empowered recommendation framework to execute multi-stage tasks (*e.g.,* recall, ranking, re-ranking) via through chain-of-recommendations. To deal with large-scale item sets, we design a novel strategy to structure all items into a hierarchical tree structure, *i.e., semantic item tree*. The semantic item tree can be dynamically updated to incorporated new items and effectively retrieved according to user interests. Extensive experiments on MIND and Amazon datasets indicate that even under the zero-shot setting, UniLLMRec achieves competitive performance compared to conventional recommendation models. Hence, UniLLMRec not only simplifies the recommendation process but reduces the reliance on large-scale training datasets as well. This enables more efficient and scalable deployment in various recommendation scenarios. In the future, we will work to improve the balance of the semantic tree to avoid model performance degradation resulting from the imbalance tree structure.

Limitations

Our proposed UniLLMRec can effectively work on large-scale item corpus in a zero-shot manner. It mainly leverages the semantic item tree, by dynamically updating its structure in real-time. Therefore, there is no need for LLM fine-tuning to cater dynamic item updates. Nevertheless, the tree is manually constructed based on item semantic information, similar items with subtle semantic differences can be assigned to different subcategories, thus introducing noises to our model. Additionally, issues such as structural imbalance and uneven item set sizes in leaf nodes exist in the constructed semantic tree. For instance, in the MIND dataset, a majority of news concentrates on a few subcategories. Similar issues arise in the Amazon dataset, exhibiting the imbalanced distribution of the constructed tree. We leave it for future work to construct a balanced semantic tree with accurate semantic information.

Ethics Statement

The proposed LLM-empowered recommendation systems leverage user interaction history to infer user interests and conduct recommendation tasks. Therefore, on the one hand, for privacy concerns, the utilization of user data for inference is contingent upon obtaining explicit user consent. On the other hand, though concerns regarding fairness in large language models may arise, our proposed model solely relies on user historical interaction data, avoiding any involvement with sensitive user attributes, thus mitigating potential fairness issues associated with these features.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

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*.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*.

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. *arXiv preprint arXiv:2305.02182*.

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 *RecSys*, pages 299–315.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to index item ids for recommendation foundation models. *arXiv preprint arXiv:2305.06569*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, et al. 2023. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*.

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

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*, pages 3–11.

Aleksandr V Petrov and Craig Macdonald. 2023. Generative sequential recommendation with gptrec. *arXiv preprint arXiv:2306.11114*.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. Personalized news recommendation with knowledge-aware interactive matching. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–70.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1161–1170.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 836–845.

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. 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 6389–6394.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*.

Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130.

Zizhuo Zhang and Bang Wang. 2023. Prompt learning for news recommendation. *arXiv preprint arXiv:2304.05263*.

A Experimental Details

A.1 Datasets

- **MIND**[3] is a large-scale news recommendation dataset collected from Microsoft News. It includes the historical behaviors of click and non-

[3] https://msnews.github.io/

click on English news articles from a million users.

- **Amazon-Review**[4] dataset is crawled from Amazon e-commerce platform containing not only the reviews and ratings from users but also the product information like price and category.

Besides, for user-item interaction sequences, we limited their length to 50. Handling the extensive item subsets from some leaf nodes posed challenges for direct input into the Language Model (LLM) using a single prompt template. Consequently, we constrained each subset to a maximum of 50 items. Positive items were grouped into their respective subsets, and negative sampling was applied to each leaf node until reaching a size of 50. This process resulted in a candidate set of 1217 items for MIND and 6176 items for Amazon.

A.2 Baseline implementation details

For popularity-based models, we recall the top 20 items with the highest historical click-through rates based on user click history. Unfortunately, none of these 20 items were successfully recalled within the 500 samples selected for evaluation. Therefore, we conducted experiments on the entire test set, and the results are reported in Figure 4. The FM, DeepFM, NRMS, and SASRec models all adopt a dual-tower structure, utilizing the Adam optimizer. Cross entropy is employed as the loss function, with a negative sampling ratio of 1 across all models. In the MIND dataset, all models use the item title as the input feature. However, in the Amazon dataset, using only the title feature leads to suboptimal performance for FM, DeepFM, NRMS, and SASRec. Consequently, for the Amazon dataset, we substitute the item description feature for the title as the input. For the DeepFM model, we set the embedding dimension to 50. In NRMS and SASRec, the embedding dimension is set to 300, with a dropout rate of 0.2.

[4] https://cseweb.ucsd.edu/ jmcauley/datasets.html