LARGE LANGUAGE MODELS AUGMENTED RATING PREDICTION IN RECOMMENDER SYSTEM

Sichun Luo[1,2], *Jiansheng Wang*[3], *Aojun Zhou*[4], *Li Ma*[5], *Linqi Song*[1,2†]

[1]Department of Computer Science, City University of Hong Kong
[2]City University of Hong Kong Shenzhen Research Institute
[3]College of Water Resources and Architectural Engineering, Northwest A&F University
[4]Department of Electronic Engineering, The Chinese University of Hong Kong
[5]Library, Tsinghua University

ABSTRACT

Recently, large language models (LLMs) have demonstrated impressive capabilities and gained widespread applications. However, their direct application to recommendation tasks (e.g., rating prediction task) often falls short of optimal results due to a lack of understanding of collaborative information in recommendations. In this paper, we propose **L**arge l**A**nguage **M**odel **A**ugmented **R**ecommendation (LAMAR) framework to address this limitation. Instead of relying solely on LLMs, our framework combines their outputs with traditional recommendation models, leveraging both collaborative and semantic information. We further enhance the recommendation performance through an ensemble of diverse prompts and utilize LLMs to extract side information for augmenting traditional recommendation models. Empirical studies on real-world datasets demonstrate that LAMAR outperforms existing approaches, highlighting the benefits of leveraging LLMs in recommendation systems. Code is available at `https://github.com/sichunluo/LAMAR`.

***Index Terms*—** recommender system, large language model, rating prediction

1. INTRODUCTION

Recommender systems have become widely adopted in various domains to address the issue of information overload [1, 2, 3]. One crucial task in recommendation is rating prediction, which involves predicting the ratings or preferences that a user would assign to items in a recommender system [4]. Traditional recommender systems often employ neural networks or similar models to generate recommendations based on user preferences [5, 6]. Although these systems have shown effectiveness, they also have inherent limitations. Specifically, two key challenges emerge: First, traditional models typically transform features into embeddings, neglecting the textual semantic information associated with the features. Second, these systems often lack side information, such as movie directors in movie rating prediction, which hinders their ability to achieve better performance.

In recent years, large language models (LLMs) have demonstrated remarkable capabilities in various domains, including natural language understanding, language generation, and complex reasoning [7, 8]. However, in contrast to computer vision and natural language processing, the field of recommender systems lacks a standardized *foundation model*, making it challenging to transfer knowledge between different recommendation scenarios [9]. Drawing inspiration from the success of LLMs in other domains, researchers have begun exploring the potential of applying LLMs to recommender systems [10, 11]. Unlike traditional recommender systems that typically rely on training neural networks to model user preferences [5], LLM-based recommendations involve directly prompting the LLMs to generate recommendations. Therefore, we aim to leverage LLMs to enhance the existing traditional recommender systems.

Nevertheless, preliminary explorations into the integration of LLMs, such as ChatGPT, into recommendation tasks have yielded less satisfactory outcomes [10, 11]. LLMs still encounter formidable challenges in various recommendation tasks, including sequential and top-k recommendations. Consequently, the direct utilization of LLMs for recommendation purposes may not be the most optimal approach. The limitations in leveraging LLMs for recommendation can be attributed to two pivotal factors. Firstly, LLMs encounter difficulties in comprehending the semantic meaning of user/item IDs, which poses challenges for ID-based recommendations. Secondly, the extensive candidate set involved in recommendation tasks makes it arduous to incorporate and fully comprehend the collaborative information within the LLMs.

In this paper, we propose **L**arge l**A**nguage **M**odel **A**ugmented **R**ecommendation (LAMAR), a general model agnostic framework that augments the traditional recommendation systems by integrating LLMs. LAMAR introduces an adaptive merging module to effectively combine these two methods, which

†Corresponding Author

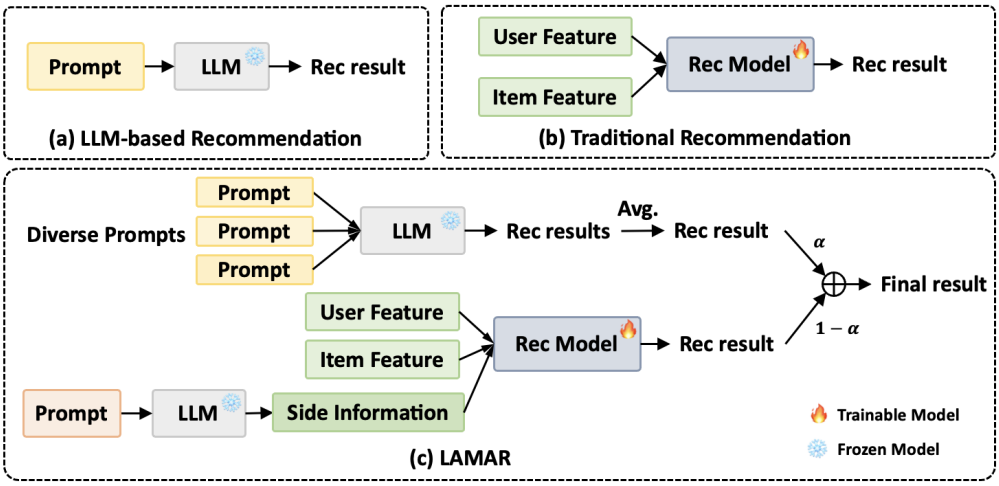

Fig. 1: The overall framework of LAMAR, with comparison with (a) LLM-based Recommendation and (b) Traditional Recommendation.

allows us to leverage the advantages of traditional recommendation models, which excel at understanding ID features and collaborative information, and LLM-based recommendation models, which excel at comprehending textual information and reasoning. By incorporating an adaptive merging module, we can optimize the performance of the recommendation system. Moreover, we employ a diverse prompt ensemble technique to generate multiple answers from LLMs. By averaging these answers, we can obtain more accurate results, taking advantage of the diverse perspectives provided by the LLMs. Furthermore, we utilize the capabilities of LLMs to provide valuable side information, which supports the traditional recommendation model in making more accurate predictions. By leveraging the additional information provided by LLMs, we enhance the accuracy and effectiveness of the recommendation system. We performed extensive experiments on two real-world datasets, employing various backbone models, to evaluate the performance of our proposed LAMAR framework in the rating prediction task. The experimental results demonstrate the effectiveness of our method in effectively augmenting existing recommendation models.

In a nutshell, our contribution is threefold.

- We introduce a novel and model-agnostic framework called LAMAR, which enhances traditional recommendation models by incorporating LLMs. By leveraging the semantic information comprehension and reasoning abilities of LLMs, our framework extends the capabilities of traditional recommendation models.
- LAMAR prompt LLMs to generate side information that augments traditional recommendation models. Furthermore, we propose using diverse prompts ensemble to further improve the recommendation performance of the LAMAR framework.
- Empirical studies conducted on the rating prediction task demonstrate the effectiveness of LAMAR. We observe obvious improvements in recommendation performance through rigorous evaluation and comparison with existing approaches.

2. METHOD

Figure 1 illustrates the architecture of our proposed LAMAR framework. In contrast to traditional recommendation models, we integrate the strengths of both traditional and LLM-based recommendation models to leverage their respective advantages. Additionally, we incorporate diverse prompts to further enhance the performance of the framework. Moreover, we prompt LLMs to generate valuable side information that enriches the traditional recommendation model.

2.1. Recomemndation with LLMs

2.1.1. Prompt Construction for Rating Prediction Task

Following [10], we construct prompts tailored to the specific characteristics of the rating prediction task. These prompts serve as inputs for the LLMs, enabling them to generate recommendation results based on the prompt specifications.

The prompt template we employ is as follows:

```
### Instruction:
Based on the rating history below, please
predict user's rating for the movie:
{movie_name}. The output must predict the
user's rating, and then explain the reasons
why the user will give the rating. While
predicting the user's rating for the movie,
the user's preference and the features of
the movie should be considered.
```

```
### Information:
Here is user rating history:
{rating_history}
### Format Example:
title: <Movie Title>
rating: x stars
reasons: {reasons}
### Answer: {LLM_output}
```

2.1.2. Diverse Prompts

Motivated by the work of [12, 13], we incorporate the use of diverse prompts to augment the performance of our LAMAR framework. By employing different prompts, we can elicit different reasoning paths within the LLMs, leading to more reliable and comprehensive results.

In our approach, the process of generating rating prediction using diverse prompts can be denoted as $r = \theta(p)$, where p represents the prompt used, and θ represents the LLM-based recommendation model. We utilize k diverse prompts, resulting in multiple recommendation outputs: $\{r_1 = \theta(p_1), \ldots, r_k = \theta(p_k)\}$. To consolidate these multiple outputs and obtain a final score, we employ an averaging mechanism. Specifically, we compute the average of the k recommendation outputs as: $r_{\text{LLM}} = \frac{1}{k}\sum_{i=1}^{k} r_i$. By averaging the diverse recommendation results, we obtain a more robust and accurate final score for the recommendations.

2.2. Side Information Augmented Recommendation

In the traditional recommendation scenario, the available data for modeling is often limited. However, aggregating side information can significantly enhance the recommendation performance. To leverage side information, we utilize LLMs within our LAMAR framework.

In the traditional recommendation model, we can denote the recommendation process as $r = \psi(f_{\text{id}}, f_{\text{feature}})$, where ψ represents the traditional recommendation model, such as DeepFM [5] or other similar models. Here, f_{id} represents the ID features, and f_{feature} represents additional features. To incorporate side information provided by LLMs, we introduce the LLM denoted as θ. By applying a prompt p' to the LLM, we obtain the side information f_{s}, which captures additional semantic knowledge related to the recommendation task. Hence, the rating prediction can be expressed as $r_{\text{Rec}} = \psi(f_{\text{id}}, f_{\text{feature}}, f_{\text{s}})$, where we augment the traditional recommendation model with the LLM-generated side information.

The prompt template we utilize within our LAMAR framework to generate side information is as follows:

```
### Instruction:
Provide detailed information of the movie
{movie_name}, including the director,
scriptwriters, stars, and keywords of the
plot and style. The provided information
must be correct.
### Format Example:
{example}
### Answer: {LLM_output}
```

2.3. Adaptive Merging

In the context of the long-tail phenomenon, where users with a large number of interactions tend to perform better in traditional recommendation models [14, 15], we propose an adaptive merging approach within our LAMAR framework. This adaptive merging combines the results obtained from both LLM-based and traditional recommendation models adaptively. The final score for user u and item v is computed as: $r_{u,v} = \alpha r^{u,v}_{\text{LLM}} + (1-\alpha) r^{u,v}_{\text{Rec}}$, where $r^{u,v}_{\text{LLM}}$ represents the rating prediction score for user u and item v from the LLM-based model, and $r^{u,v}_{\text{Rec}}$ represents the score from the traditional recommendation model. The hyperparameter α controls the weight given to each recommendation model's score. To adaptively determine α, we consider the number of interactions for user u. If the number of interactions exceeds a threshold γ, we set $\alpha = \alpha_1$. Otherwise, we set $\alpha = \alpha_2$, where $\alpha_1 < \alpha_2$.

This adaptive merging mechanism allows us to dynamically adjust the contribution of the LLM-based model and the traditional recommendation model based on the user's interaction history.

3. EXPERIMENT

In this section, we perform experiments on real-world datasets for evaluating various methods on rating prediction tasks.

Table 1: Dataset Statistics.

Dataset	Kaggle-Movie	MovieLens-100K
# of user	670	943
# of item	5,977	1,682
# of rating	96,761	100,000
density	0.024162	0.063046
User Features	N/A	IDs, Gender, Occupation ZipCode, Age
Item Features	IDs, Title, Genres, Year	
LLM Generated Features	Movie Director, Scriptwriter, Stars	

3.1. Experiment Setup

Dataset. To evaluate the effectiveness of proposed LAMAR, we conduct experiments on MovieLens-100K (ML-100K) [16] dataset, which is widely used for evaluating recommendation algorithms in the context of movie rating prediction. We also use Kaggle-Movie [17], which is an extended MovieLens dataset released on Kaggle. The characteristics of datasets are summarized in Table 1.

Table 2: Performance achieved by different methods. The better results are highlighted in **boldfaces**.

Backbone	w/ LAMAR	Kaggle Movie		ML-100K	
		RMSE ↓	MAE ↓	RMSE ↓	MAE ↓
LLaMA	–	1.0404	0.7459	1.1886	0.9093
DeepFM	×	0.9873	0.7765	1.1204	0.8729
	✓	**0.9582**	**0.7505**	**1.0778**	**0.8399**
NFM	×	1.0379	0.7806	1.0581	0.8323
	✓	**0.9882**	**0.7500**	**1.0345**	**0.8157**
DCN	×	1.0299	0.7794	1.1121	0.8591
	✓	**0.9907**	**0.7531**	**1.0765**	**0.8343**
AFM	×	1.0378	0.8016	1.0938	0.8385
	✓	**0.9980**	**0.7722**	**1.0623**	**0.8203**
xDeepFM	×	1.0625	0.8134	1.2528	0.9766
	✓	**1.0094**	**0.7712**	**1.1497**	**0.9007**
AutoInt	×	0.9881	0.7636	1.0970	0.8581
	✓	**0.9646**	**0.7423**	**1.0613**	**0.8296**

Table 3: Ablation study on ML-100K dataset with backbone model DeepFM. The best results are highlighted in **boldfaces**.

Variants	RMSE ↓	MAE ↓
LAMAR	**1.0778**	**0.8399**
w/o diverse prompt	1.0952	0.8547
w/o side information	1.0825	0.8438
w/o adaptive merging	1.0958	0.8541

Evaluation Metrics. In line with [18], we employ two evaluation metrics: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

Backbone Models. We incorporate our LAMAR with the following recommendation models that are often used for predicting ratings in recommender systems: CCPM [19], DeepFM [5], NFM [1], DCN [20], AFM [21], xDeepFM [22], and AutoInt [23]. To make our results fully reproductive, we utilize the open-source LLaMA [8] model as the backbone model for LLM.

Experiment Settings. To ensure a fair comparison, we adopt the best hyper-parameter settings reported in the original papers of the baselines, and fine-tune all baseline hyper-parameters using grid search. We adopt the leave-one-out strategy [17] to use the last interacted item to test, the second last interacted item to validate, and others to train for each user. In the default setting, α_1 is set to 0.1, α_2 is set to 0.3, and γ is set to 80.

3.2. Main Result

We evaluate baselines and our method using two datasets to evaluate the performance under different scenarios. Table 2 summarizes the model performance on these datasets. Through our experiments, we observed that by leveraging LLMs to augment traditional recommendation models, LAMAR significantly improves the recommendation performance. The collaborative information captured by traditional models, combined with the semantic information provided by LLMs, leads to more accurate rating prediction. Besides, comparing the performance of LAMAR against standalone LLM models, we observed that LAMAR consistently outperforms LLM models in terms of recommendation accuracy. This suggests that the integration of LLMs within the traditional recommendation framework enhances the overall performance, leveraging the strengths of both approaches.

3.3. Ablation Study

We perform ablation studies to analyze different components of our model. The results are shown in Table 3. Specifically, we build three variants: w/o diverse expert, w/o side information, and w/o adaptive merging. By using multiple prompts, we enable the LLMs to capture a broader range of semantics, resulting in more comprehensive and accurate recommendations. In addition, through adaptive merging method, LAMAR achieved superior recommendation performance compared to using standard ones. Moreover, side information merging improves recommendation quality. The utilization of LLMs to extract side information that augments traditional recommendation models proved to be highly effective. This demonstrates the value of leveraging LLMs for extracting relevant and complementary information to enhance recommendations.

4. CONCLUSION

In this paper, we introduces the LAMAR framework, which leverages LLMs to augment recommendation systems. We address the limitation of LLMs in understanding collaborative information by combining their outputs with traditional recommendation models. By incorporating both the collaborative information extracted by traditional models and the semantic information extracted by LLMs, LAMAR achieves improved recommendation performance. We further enhance LLM-based recommendations by employing an ensemble of diverse prompts, which boosts the effectiveness of the framework. Additionally, we utilize LLMs to extract side information that enhances traditional recommendation models, providing a comprehensive approach to recommendation tasks. Empirical studies conducted on real-world datasets validate the effectiveness of our proposed method.

5. ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 62371411, the Research Grants Council of the Hong Kong SAR under Grant GRF 11217823, InnoHK initiative, the Government of the HKSAR, Laboratory for AI-Powered Financial Technologies.

6. REFERENCES

[1] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[2] Sichun Luo, Yuanzhang Xiao, and Linqi Song, "Personalized federated recommendation via joint representation learning, user clustering, and model adaptation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4289–4293.

[3] Sichun Luo, Yuanzhang Xiao, Xinyi Zhang, Yang Liu, Wenbo Ding, and Linqi Song, "Perfedrec++: Enhancing personalized federated recommendation with self-supervised pre-training," *arXiv preprint arXiv:2305.06622*, 2023.

[4] Zahid Younas Khan, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince, "Deep learning techniques for rating prediction: a survey of the state-of-the-art," *Artificial Intelligence Review*, vol. 54, pp. 95–135, 2021.

[5] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[6] Sichun Luo, Chen Ma, Yuanzhang Xiao, and Linqi Song, "Improving long-tail item recommendation with graph augmentation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1707–1716.

[7] OpenAI, "Gpt-4 technical report," 2023.

[8] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[9] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

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

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

[12] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen, "Making language models better reasoners with step-aware verifier," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 5315–5333.

[13] Sinuo Deng, Lifang Wu, Ge Shi, Lehao Xing, Meng Jian, and Ye Xiang, "Learning to compose diversified prompts for image emotion classification," *arXiv preprint arXiv:2201.10963*, 2022.

[14] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen, "Challenging the long tail recommendation," *arXiv preprint arXiv:1205.6700*, 2012.

[15] Elaheh Malekzadeh Hamedani and Marjan Kaedi, "Recommending the long tail items through personalized diversification," *Knowledge-Based Systems*, vol. 164, pp. 348–357, 2019.

[16] F Maxwell Harper and Joseph A Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[17] Sichun Luo, Xinyi Zhang, Yuanzhang Xiao, and Linqi Song, "Hysage: A hybrid static and adaptive graph embedding network for context-drifting recommendations," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1389–1398.

[18] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[19] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang, "A convolutional click prediction model," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 1743–1746.

[20] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, pp. 1–7. 2017.

[21] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," *arXiv preprint arXiv:1708.04617*, 2017.

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

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