LLMs for Recommendation

ZHENG Yuhui

I. Related Work

Recommendation systems play a critical role in assisting users in finding relevant and personalized items or content. With the emergence of Large Language Models (LLMs) in Natural Language Processing (NLP), there has been increasing interest in leveraging the capabilities of these models to enhance recommendation systems [1]. The primary advantage of integrating LLMs into recommendation systems lies in their ability to tap into the vast external knowledge encoded within them. These models can generalize to previously unseen candidates, thanks to extensive pre-training on factual information, domain expertise, and common-sense reasoning. This allows LLMs to provide relevant recommendations even without prior exposure to specific items or users. Furthermore, the interpretability of recommendations is enhanced, as LLM-based systems can offer explanations derived from their language generation capabilities [2], thereby helping users understand the factors influencing the recommendations. The application of LLMs in recommendation systems primarily falls into two categories: deep data representation using LLMs, and the direct application of generative LLMs to construct recommendation logic.

In deep representation, discriminative language models, such as BERT, are commonly used for fine-tuning and pre-training, where they integrate domain-specific data features to improve the performance of recommendation systems. For instance, U-BERT [3] leverages content-rich domain data to learn user representations, thereby compensating for the lack of sufficient behavioral data. Similarly, UserBERT [4] incorporates two self-supervised tasks for pretraining on unlabeled behavioral data.

Recent studies have highlighted the significant potential of generative LLMs in recommendation systems through prompting and tuning techniques. Notable advancements include: Liu et al. [5] conducting a comprehensive evaluation of ChatGPT's performance across five key recommendation tasks. Sanner et al. [6] designed three distinct prompt templates to assess the effectiveness of prompts, finding that zero-shot and few-shot strategies are particularly effective for preference-based recommendations using language. Sileo et al. [7] and Hou et al. [8] focused on developing tailored prompt methods for specific recommendation tasks. Gao et al. [2] introduced ChatREC, an interactive recommendation framework based on ChatGPT, which refines user preferences through multiple rounds of dialogue. Kang et al. [9] explored formatting user historical interactions as prompts, evaluating the performance of LLMs at various scales. Dai et al. [10] designed templates for diverse recommendation tasks using demonstration examples. Bao et al. [11] developed TALLRec, which showcases the potential of LLMs in recommendation systems through two-stage fine-tuning. Ji et al. [12] proposed GenRec, a method that leverages the generative capabilities of LLMs to directly produce recommendation targets. In specific applications, such as online recruitment, generative models like GIRL [13] and reclm [14] have demonstrated enhanced explainability and appropriateness in recommendations.

II. Preliminaries

A. Generative Large Language Models

Generative Large Language Models (LLMs) are a type of model based on transformer decoders designed to generate natural language text. Trained on massive text corpora, LLMs capture broad contextual relationships between words. These models generate a sequence of words $(w_1, w_2, ..., w_n)$ by modeling the joint probability of word sequences, $P(w_1, w_2, ..., w_n)$.

B. Prompt

Prompts are designed to direct generative large language models to produce specific responses. They act as guides, steering the model to generate text tokens within particular contexts or styles [15].

C. Chain-of-Thought Prompting

Chain of Thought (CoT) is a method of thinking designed to help people or LLMs reason and solve problems in a more systematic way by breaking down complex problems into a series of ordered thinking steps. The core of the method is the use of intermediate reasoning steps to make the final answer clearer and more accurate.

Chain of Thought Prompting refers to the practice of adding guiding phrases, such as `reasoning step by step`, `providing explanations for your answers`, or breaking down complex questions into smaller sub-questions, to guide large language models to generate responses in a chain of thought format. This method aims to improve the coherence and correctness of the generated responses.

III. Methods

The project aims to provide meal recommendations based on the staff's code workload and preferences, which requires the recommendation system to have both code workload evaluation and commonsense reasoning abilities. Based on the above survey, we know that some open-source SOTA generative LLMs possess strong commonsense reasoning capabilities. Additionally, due to the large amount of code data used during the training of generative LLMs, they also have sufficient code workload evaluation abilities. Finally, due to computational resource limitations, I have chosen a recommendation system architecture based on a SOTA generative LLM combined with CoT Prompting.

Prompt

```
You are a professor in the Computer Science Department at Carnegie
Mellon University, and at the same time, you are an experienced
nutritionist. You will be provided with a piece of code, a menu, and
the dining requirements of staff members. You have the following two
tasks:
1. Assess the energy expenditure of the staff members based on the
complexity and workload of the code.
2. Combine the energy expenditure and the dining requirements of the
staff members to select and match a nutritious meal from the menu that
meets their dining requirements. Greater energy expenditure
necessitates more nutritious food.
Please answer the staff's energy consumption and the reasons.
Additionally, elaborate on the reasons for selecting and matching the
meal.

Now, the code is
```
<CODE>
```
, the menu is "<MENU>" and the dining requirements of staff members
are "<REQs>". Please answer as requested above.
```

Fig. 1. **Chain of Thought Prompting**.

A. Model Selection

I utilise opensource SOTA LLMs by the Huggingface open source community[1] APIs. Huggingface has many excellent open-source generative LLMs, but based on the task's requirements for code capabilities and commonsense reasoning, I referred to the CompassBench Large Language Model Leaderboard[2] and chose Qwen2.5-72B-Instruct as the backbone.

B. Prompt Engineering

To further improve the rationality, interpretability and accuracy of the recommendation system, I have divided the recommendation task into two sub-tasks: 1. First, evaluate the code workload; 2. Then, make recommendations based on the workload and the staff's requirement, as shown in the red fields of Figure 1. Additionally, I prompted LLM to provide reasons for the decisions, as indicated in the green fields of Figure 1. Figure 2 is a recommended sample.

References

[1] P. Liu, L. Zhang, and J. A. Gulla, "Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems," 2023. [Online]. Available: https://arxiv.org/abs/2302.03735

[2] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-rec: Towards interactive and explainable llms-augmented recommender system," 2023. [Online]. Available: https://arxiv.org/abs/2303.14524

[3] Z. Qiu, X. Wu, J. Gao, and W. Fan, "U-bert: Pre-training user representations for improved recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4320–4327.

[4] C. Wu, F. Wu, Y. Yu, T. Qi, Y. Huang, and X. Xie, "Userbert: Contrastive user model pre-training," 2021. [Online]. Available: https://arxiv.org/abs/2109.01274

[5] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, and Y. Zhang, "Is chatgpt a good recommender? a preliminary study," 2023. [Online]. Available: https://arxiv.org/abs/2304.10149

[6] S. Sanner, K. Balog, F. Radlinski, B. Wedin, and L. Dixon, "Large language models are competitive near cold-start recommenders for language- and item-based preferences," 2023. [Online]. Available: https://arxiv.org/abs/2307.14225

[1]https://huggingface.co/docs/api-inference/supported-models

[2]https://rank.opencompass.org.cn/leaderboard-llm/?m=24-09

[7] D. Sileo, W. Vossen, and R. Raymaekers, "Zero-shot recommendation as language modeling," 2021. [Online]. Available: https://arxiv.org/abs/2112.04184

[8] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao, "Large language models are zero-shot rankers for recommender systems," 2024. [Online]. Available: https://arxiv.org/abs/2305.08845

[9] W.-C. Kang, J. Ni, N. Mehta, M. Sathiamoorthy, L. Hong, E. Chi, and D. Z. Cheng, "Do llms understand user preferences? evaluating llms on user rating prediction," 2023. [Online]. Available: https://arxiv.org/abs/2305.06474

[10] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, and J. Xu, "Uncovering chatgpt's capabilities in recommender systems," in *Proceedings of the 17th ACM Conference on Recommender Systems*, ser. RecSys '23. ACM, Sep. 2023, p. 1126–1132. [Online]. Available: http://dx.doi.org/10.1145/3604915.3610646

[11] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "Tallrec: An effective and efficient tuning framework to align large language model with recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, ser. RecSys '23. ACM, Sep. 2023, p. 1007–1014. [Online]. Available: http://dx.doi.org/10.1145/3604915.3608857

[12] J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, and Y. Zhang, "Genrec: Large language model for generative recommendation," 2023. [Online]. Available: https://arxiv.org/abs/2307.00457

[13] Z. Zheng, Z. Qiu, X. Hu, L. Wu, H. Zhu, and H. Xiong, "Generative job recommendations with large language model," 2023. [Online]. Available: https://arxiv.org/abs/2307.02157

[14] L. Friedman, S. Ahuja, D. Allen, Z. Tan, H. Sidahmed, C. Long, J. Xie, G. Schubiner, A. Patel, H. Lara, B. Chu, Z. Chen, and M. Tiwari, "Leveraging large language models in conversational recommender systems," 2023. [Online]. Available: https://arxiv.org/abs/2305.07961

[15] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "Gpt understands, too," 2023. [Online]. Available: https://arxiv.org/abs/2103.10385

Student Bio

ZHENG Yuhui I'm ZHENG Yuhui, and I am currently pursuing my Master's degree in the Department of Electrical Engineering, City University of Hong Kong. At the same time, I am currently working as a Research Assistant at MMLab@CUHK, advised by Prof. Xiangyu Yue and doing a remote internship at Shanghai AI Lab, advised by Prof. Yan Teng. I also interned at Beijing Academy of Artificial Intelligence, International Digital Economy Academy. My research interests are in Multimodel Learning, Reasoning, LLMs.

I was mainly responsible for the recommendation module in the project, the writing of the recommendation section of the report, and I was also the main participant in the requirements engineering. Through this process, I understood how to design good requirements, learned about the cutting-edge progress of large language models for recommendation, mastered the use of Huggingface API, and deepened my knowledge and understanding of agile software development and further mastered the GitHub software development process.

In terms of contribution, I think requirements engineering is an important part of the software development process. A good requirement determines the efficiency of software development and the value of software. Meanwhile, recommendation module is an indispensable and important part of our project.

Appendix

Code

```
import re
text = "This is <CODE> and <SESE>."
text = re.sub(r"<CODE>", "code", text)
text = re.sub(r"<SESE>", "sese", text)
print(text)
```

User Preference

I don't like chilli.

Menu

1. Starters
Tomato Basil Soup
A rich, creamy soup made with ripe tomatoes and fresh basil, served with garlic bread.
$5.99

Crispy Calamari
Golden-fried calamari rings with a zesty lemon aioli dip.
$7.99

Bruschetta Trio
Three types of bruschetta: classic tomato, mushroom & garlic, and avocado spread.
$6.99

2. Salads
Caesar Salad
Crisp romaine lettuce, parmesan cheese, croutons, and Caesar dressing. Add grilled chicken for $2.
$8.99

Mediterranean Salad
Mixed greens, cherry tomatoes, olives, feta cheese, and balsamic vinaigrette.
$9.49

3. Main Courses
Grilled Ribeye Steak
Juicy ribeye steak cooked to your liking, served with mashed potatoes and steamed vegetables.
$22.99

Herb-Crusted Salmon
Fresh salmon fillet baked with an herb crust, served with quinoa and asparagus.
$18.99

Vegetarian Pasta Primavera
Pasta tossed with seasonal vegetables in a light garlic and olive oil sauce.
$14.99

4. Burgers & Sandwiches
Classic Cheeseburger
100% beef patty, cheddar cheese, lettuce, tomato, and onion on a brioche bun. Served with fries.
$12.99

Grilled Chicken Club
Grilled chicken breast, crispy bacon, lettuce, tomato, and mayo on sourdough bread.
$13.49

5. Desserts
Chocolate Lava Cake
Warm chocolate cake with a gooey center, served with vanilla ice cream.
$6.99

New York Cheesecake
Classic cheesecake with a graham cracker crust, topped with fresh berries.
$5.99

6. Beverages
Freshly Brewed Coffee
Regular or decaf, served hot or iced.
$2.99

Lemon Mint Iced Tea
Refreshing house-made iced tea with lemon and mint.
$3.49

Soft Drinks
Coke, Sprite, Diet Coke, or Fanta.
$2.49

Qwen2.5-72B-Instruct

Energy Expenditure Assessment

The provided code is relatively simple. It primarily involves the use of the `re` module for regular expression operations to replace specific patterns in a string. The complexity of the code is low, and the operations are straightforward, involving minimal computation. Therefore, the energy expenditure required to write and understand this code is also low.

Staff Member's Energy Consumption and Meal Selection

Given that the code is not very complex, the energy expenditure for the staff member is on the lower end of the spectrum. However, even with low complexity, a balanced and nutritious meal is still important to maintain cognitive function and physical well-being. The staff member's dietary preference is to avoid chili.

Selected Meal

Main Course: Herb-Crusted Salmon
- **Reason:** This dish is a balanced and nutritious option. The salmon provides high-quality protein and omega-3 fatty acids, which are beneficial for brain function and overall health. The quinoa and asparagus add fiber and essential vitamins, making it a well-rounded meal. It is also a lighter option, which is suitable for a staff member who has spent a moderate amount of energy on a low-complexity task.

Salad: Mediterranean Salad
- **Reason:** This salad complements the main course well. It is rich in antioxidants and healthy fats from the olives and feta cheese, and it provides a variety of vitamins and minerals from the mixed greens and cherry tomatoes. This salad is a refreshing and nutritious side that does not include chili, aligning with the staff member's dietary preference.

Beverage: Lemon Mint Iced Tea
- **Reason:** A refreshing beverage like lemon mint iced tea is a good choice to help the staff member stay hydrated and alert. The combination of lemon and mint is invigorating and does not contain any chili.

Summary

- **Energy Expenditure:** Low to moderate
- **Meal Selection:** Herb-Crusted Salmon, Mediterranean Salad, and Lemon Mint Iced Tea
- **Reasons:** The selected meal is balanced, nutritious, and aligned with the staff member's dietary preferences. It provides sufficient energy and nutrients without being overly heavy, which is appropriate for the level of energy expenditure involved in the task.

Fig. 2. A recommended sample.