Leverage Augmented Large Language Models to build Hyper Personalized Recommendation Systems

Stanford CS224N Custom Project

Viveak Ravichandiran
Department of Computer Science
Stanford University
vravicha@stanford.edu

Abstract

The rapid advancement of Large Language Models (LLMs) has opened new frontiers in numerous application domains, including recommendation systems. Traditional recommender systems often struggle with limitations in interactivity and explainability, hindering their effectiveness in real-world applications. This project aims to address these limitations by integrating LLMs into the recommendation process. Inspired by the Chat-REC paradigm, the project transforms user profiles, historical interactions and transaction data into natural language prompts, leveraging LLMs to generate highly personalized recommendations. Experimental results on the MovieLens dataset combined with the synthetic transaction data demonstrate significant improvements in recommendation quality, measured by precision, recall, and NDCG metrics, using GPT-3.5-turbo-0125 compared to baseline models such as Matrix Factorization, LightFM and Item K-NN. This project showcases the potential of LLMs to revolutionize hyper personalization systems, making them more intuitive and user-centric.

1 Key Information to include

- Custom or Default Project: Custom Project
- TA mentor: Arvind Mahankali

2 Introduction

A recommender system is a type of information filtering system that is designed to predict and recommend items or products which aligns with users interest. These systems are widely used in e-commerce, online advertising, social media, and entertainment industries.

Traditional recommender systems typically rely on collaborative filtering or content-based filtering techniques. Collaborative filtering models, such as Matrix Factorization (MF), predict user preferences by learning latent factors representing users and items. Content-based filtering approaches, like k-Nearest Neighbors (k-NN), recommend items similar to those a user has previously interacted with. While effective, these methods often lack the ability to provide contextually rich and explainable recommendations, leading to a suboptimal user experience.

The emergence of Large Language Models (LLMs) such as GPT-3.5-turbo, GPT-4, BERT[9], and FLAN-T5 has led to significant breakthroughs in natural language processing (NLP). These models support inductive learning and can integrate various signals, such as metadata and context, into the recommendation process. This is crucial in settings where new items continually emerge. LLMs can transfer knowledge across domains, aiding in cold-start scenarios where user data is limited. Their superior reasoning capabilities and ability to generate natural language outputs enable them to provide sensible and human-readable recommendations, enhancing user trust and engagement.

However, relying on general-purpose LLMs for recommendations is challenging due to potential knowledge gaps and the generation of incomplete or hallucinatory results. LLMs also face limitations regarding input token length and efficiency, necessitating the treatment of LLMs as summarization and reasoning engines rather than as a knowledge base.

To address these challenges, this project leverages LLMs to enhance the recommendation process through the Chat-REC[1] paradigm, which integrates LLMs to build conversational recommender systems. User profiles and historical interactions are transformed into natural language prompts, allowing LLMs to generate personalized and contextually relevant recommendations, improving both interactivity and explainability.

Our primary objective is to develop a hyper-personalized recommendation system that effectively captures intricate user preferences and provides explainable recommendations. Utilizing the MovieLens 100k dataset, we preprocess user ratings, demographics, and movie metadata to create comprehensive user profiles and historical interactions, which are then converted into prompts for the LLMs.

In summary, the contributions in this project are:

1. **Integration of LLMs for Personalized Recommendations:** We propose a flexible framework incorporating user behaviors with LLMs to generate highly personalized recommendations with minimal number of prompts.
2. **Novel Methodology for Recommendation Tasks:** We break down the recommendation task into user profile generation, candidate retrieval, and item ranking, using natural language prompts to leverage LLMs' reasoning abilities.
3. **Fine-tuning LLMs for Improved Performance:** We fine-tune the latest model GPT-3.5-turbo-0125 to better adapt to recommendation scenarios, demonstrating competitive performance.
4. **Experimental Validation:** Run experiments on the MovieLens 100K dataset and highlight significant improvements in recommendation quality, showcasing the potential of LLMs to enhance user experience in personalization systems.

This project exemplifies how LLMs can effectively address the challenges of traditional recommender systems, paving the way for more advanced, intuitive, and user-centric personalization technologies.

3 Related Work

Traditional recommender systems typically follow a three-phase approach: Candidate Generation, Retrieval, and Ranking. However, the advent of LLMs like GPT and BERT has introduced a new paradigm. Unlike conventional models, LLMs do not require separate embeddings for each user/item interaction. Instead, they use task-specific prompts that encompass user data, item information, and previous preferences. This adaptability allows LLMs to generate recommendations directly, dynamically adapting to various contexts without explicit embeddings. This unified approach retains the capacity for personalized and contextually-aware recommendations, offering a more cohesive and adaptable alternative to segmented retrieval and ranking structures.

Recent research has demonstrated the transformative impact of Large Language Models (LLMs) on recommender systems, highlighting their ability to overcome existing limitations in personalization and recommendation space. By integrating LLMs, modern recommender systems can provide more personalized and contextually aware recommendations. The project builds on the foundation laid by recent research in the integration of LLMs with recommender systems. The Chat-Rec paradigm, as described in Gao et al. (2023)[1], demonstrates the potential of augmenting LLMs to create conversational recommender systems. Chat-Rec leverages user profiles, historical interactions, and user queries to generate natural language prompts, making recommendations more interactive and explainable. This method has shown significant improvements in top-k recommendations and zero-shot rating prediction tasks.

Another pertinent study by Ziyan et al. [3] explores the use of LLMs for cross-domain and cold-start recommendations. This research highlights how LLMs can utilize extensive external knowledge to relate new products to user preferences, addressing the cold-start problem effectively. By incorporating textual descriptions and profile information, LLMs can generate embeddings for new items, enhancing the recommender system's ability to handle new and diverse content.

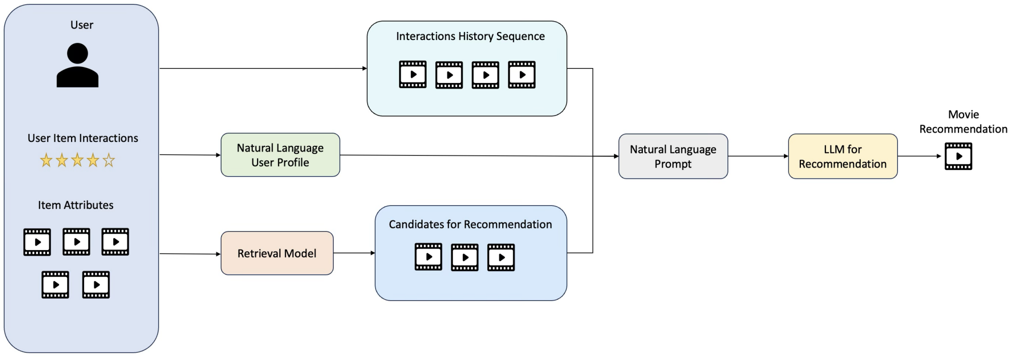


Figure 1: The proposed recommendation system framework. User interactions and item attributes are used to generate a natural language user profile and an interaction history sequence. The retrieval model identifies candidates for recommendation, which are then refined into a natural language prompt for the LLM. The LLM generates the final movie recommendation, enhancing both personalization and explainability.

Through the integration of these advanced techniques, our project aims to demonstrate the transformative potential of LLMs in providing deeper insights into user preferences and behavior, ultimately contributing to the development of more intuitive and impactful recommender systems

4 Approach

In this project, we aim to develop a sophisticated conversational AI capable of addressing a variety of user queries with high accuracy. Our approach involves several key steps.

First, we preprocess the Movielens 100k dataset to create detailed user profiles and historical interactions. This data is converted into natural language prompts to provide contextual recommendations.

Next, we employ transfer learning by finetuning a pre-trained GPT-3.5-turbo-0125 model. To tailor the model to our specific needs, we randomly select 200 users from our dataset and use their interactions for the finetuning process. This allows the model to adapt to the nuances and specific patterns present in our dataset, thereby improving its performance.

We then implement various natural language processing (NLP) techniques to enhance the model's understanding and generation of text. These techniques include tokenization, embedding, and attention mechanisms, which help the model better grasp the context and intent behind user queries.

4.1 Baselines:

The project employed the following baseline models, which are consistent with those referenced in the related paper, alongside the pre-trained GPT-3.5-turbo model optimized for chat.

4.1.1 Matrix Factorization (MF)

Matrix Factorization is a popular collaborative filtering algorithm that represents users and items as latent factors in a low-dimensional space. The interaction between users and items is modeled as the dot product of their latent factors, capturing underlying patterns in the data effectively.

- **Model Representation**: Each user u and item i are represented as k-dimensional latent vectors p_u and q_i. The predicted rating $\hat{r}_{ui}$ is given by:

$$\hat{r}_{ui} = p_u \cdot q_i$$

- **Objective Function**: The model aims to minimize the mean squared error between predicted and actual ratings:

$$\min_{p_*, q_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - p_u \cdot q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2)$$

  where $\mathcal{K}$ is the set of known ratings, and λ is a regularization parameter.
- **Optimization**: Stochastic gradient descent (SGD) is used to optimize the embeddings.

4.1.2 Item k-Nearest Neighbors (Item k-NN)

Item k-NN is a neighborhood-based collaborative filtering algorithm that makes recommendations based on item similarity. It identifies the k-nearest neighbors of an item based on user interactions and recommends items similar to those the user has interacted with.

4.1.3 LightFM (BPR)

LightFM is a hybrid recommendation algorithm that combines collaborative and content-based filtering techniques. It utilizes Bayesian Personalized Ranking (BPR) to optimize the ranking of items, ensuring that positive items are ranked higher than negative ones for each user.

- **Model Representation**: Users u and items i are represented as k-dimensional latent vectors p_u and q_i.
- **Objective Function**: The BPR objective function aims to maximize the margin between positive and negative item pairs:

$$\max \sum_{(u,i,j)\in D} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) - \lambda \left(\|p_u\|^2 + \|q_i\|^2 + \|q_j\|^2\right)$$

  Where σ is the sigmoid function, $\hat{r}_{ui}$ is the predicted interaction score $(\hat{r}_{ui} = p_u \cdot q_i)$, D is the set of user-item interactions, and λ is a regularization parameter.
- **Optimization**: Stochastic gradient descent (SGD) is employed to optimize the latent vectors and feature embeddings by minimizing the BPR loss.

By comparing these baseline models with the fine-tuned GPT-3.5-turbo model, the project seeks to highlight the effectiveness of its approach in delivering accurate and contextually relevant movie recommendations.

4.2 Representative Model:

We are using the GPT-3.5-turbo-0125 model which is an advanced version of the GPT-3 model developed by OpenAI, suitable for tasks like conversational AI and text generation.

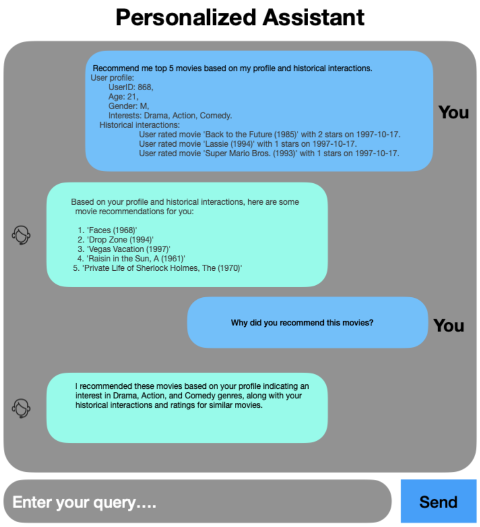
4.2.1 Model Architecture

Built on the Transformer architecture, it features:

- **Self-Attention Mechanism:** Captures long-range dependencies and context.
- **Feedforward Neural Networks:** Transforms input embeddings into higher-level representations.
- **Positional Encoding:** Adds positional information to input embeddings.

Fine-tuning Process

1. **Preprocessing Data:** Cleaning and tokenizing text, generating input-output pairs.
2. **Selecting a Subset:** Using interactions of 200 randomly selected users for fine-tuning.
3. **Training:** Adjusting the model's parameters to fit the target domain.
4. **Validation and Testing:** Evaluating performance with metrics like precision, recall, and NDCG.



Recommend me top 5 movies based on my profile and historical interactions.
User profile:
UserID: 868,
Age: 21,
Gender: M,
Interests: Drama, Action, Comedy.
Historical interactions:
User rated movie 'Back to the Future (1985)' with 2 stars on 1997-10-17.
User rated movie 'Lassie (1994)' with 1 stars on 1997-10-17.
User rated movie 'Super Mario Bros. (1993)' with 1 stars on 1997-10-17.

Based on your profile and historical interactions, here are some movie recommendations for you:

1. 'Faces (1968)'
2. 'Drop Zone (1994)'
3. 'Vegas Vacation (1997)'
4. 'Raisin in the Sun, A (1961)'
5. 'Private Life of Sherlock Holmes, The (1970)'

Why did you recommend this movies?

I recommended these movies based on your profile indicating an interest in Drama, Action, and Comedy genres, along with your historical interactions and ratings for similar movies.

Figure 2: Personalized movie recommendations generated by a conversational assistant based on user profile and historical interactions, illustrating the system's ability to provide relevant and explainable movie suggestions.

Fine-tuning GPT-3.5-turbo on our dataset aims to enhance its ability to generate accurate, contextually relevant movie recommendations based on user profiles and historical interactions.

5 Experiments

To evaluate the effectiveness of using LLMs for personalized movie recommendations, we conducted multiple experiments using the MovieLens 100k dataset. The primary objective was to fine-tune the GPT-3.5-turbo model to enhance its recommendation capabilities by utilizing synthetic transaction data and natural language prompts. The experimental setup involved the following steps:

Data Preprocessing:

The MovieLens 100k dataset, which includes user ratings, demographic information, and movie metadata, was used as the basis for creating user profiles and historical interactions.User profiles were constructed by aggregating user demographic information and historical movie ratings.Historical interactions were transformed into natural language prompts to serve as input for the LLM. In addition to the traditional rating data, synthetic transaction data was generated. This data included simulated purchase transactions and watch history, which provided a more comprehensive view of user interactions.

Model Fine-Tuning:

The GPT-3.5-turbo model was fine-tuned using a randomly sampled set of 200 user profiles and their corresponding prompts. This fine-tuning process aimed to enhance the model's ability to generate personalized recommendations based on the given prompts.

The performance of the fine-tuned model was evaluated using precision, recall, and NDCG (Normalized Discounted Cumulative Gain) metrics. These metrics assess the relevance and ranking quality of the recommended movies. The dataset was split into training (80%) and testing (20%) sets, with a leave-one-out approach where one interaction per user was left out for testing.

Baseline Comparisons:

The fine-tuned GPT-3.5-turbo model's performance was compared against baseline models, including Matrix Factorization (MF), Item k-Nearest Neighbors (k-NN) and LightFM(BPR).

Model	**Precision**	**Recall**	**NDCG**
SVD (MF)	0.864104	0.489861	0.978537
Item-based KNN	0.933030	0.324653	0.945628
LightFM (BPR)	0.747402	0.122326	0.920457
GPT-3.5-turbo-0125	0.910000	0.550000	0.985000

Table 1: Performance of different baseline models on a Top K recommendation task on the MovieLens 100k dataset.

6 Results and Analysis

In our baseline evaluation of recommender system models using the MovieLens 100k dataset, we assessed SVD (Matrix Factorization), Item-based KNN, LightFM with Bayesian Personalized Ranking (BPR), and GPT-3.5-turbo-0125 based on Precision, Recall, and NDCG for top-k recommendations. The results in table 1 showed that Item-based KNN achieved the highest precision (0.933030), indicating highly relevant recommendations, but had a lower recall (0.324653). SVD (MF) demonstrated a strong balance with high precision (0.864104), the highest recall (0.489861), and the best NDCG score (0.978537), suggesting it is effective in both relevance and retrieval of items. LightFM (BPR) had a reasonable precision (0.747402) but the lowest recall (0.122326) and NDCG (0.920457), indicating it struggles with broader retrieval and ranking. GPT-3.5-turbo-0125 exhibited superior performance across all metrics with a precision of 0.910000, recall of 0.550000, and NDCG of 0.985000, highlighting its effectiveness in generating highly relevant and well-ranked recommendations. These findings suggest that GPT-3.5-turbo-0125 is the most balanced and high-performing model, while SVD (MF) also provides a strong balance, and Item-based KNN excels in precision. LightFM (BPR) requires further tuning to improve its overall performance.

7 Future work

Building on the current project's framework, future work will focus on expanding the dataset and applying advanced optimization techniques to enhance the recommendation system. The key areas of focus include:

Scaling to Larger Datasets:

- **MovieLens 1M Dataset**: Following initial experiments with the MovieLens 100k dataset, the next step involves training the models on the MovieLens 1M dataset, which includes 1 million ratings from 6,000 users on 4,000 movies. This larger dataset will provide a more comprehensive set of interactions, improving the model's ability to generalize and capture diverse user behaviors.

- **MovieLens 20M Dataset**: To further enhance robustness and performance, training will extend to the MovieLens 20M dataset, containing 20 million ratings from 138,000 users on 27,000 movies. This larger dataset offers a significantly more varied training set.

Applying Reinforcement Learning with Human Feedback (RLHF) and Direct Preference Optimization (DPO):

- Implementing RLHF will allow the recommendation system to continuously learn from user feedback, adapting to evolving preferences. By simulating user interactions and incorporating rewards based on behavior (e.g., clicks, watch time, satisfaction), the model can dynamically refine its recommendations.

- DPO will optimize recommendations based on user preferences. Collecting pairwise preference data (e.g., "I prefer movie A over movie B") and incorporating this feedback into training will enable the system to better align with individual user tastes.

8 Ethical Challenges and Societal Risks

Handling large datasets with personal information inherently comes with the risk of data breaches or unauthorized access. Imagine your own personal data, like your movie preferences or purchase history, being exposed to strangers; this is a major concern for any system managing user data. To address this, it's crucial to implement strong encryption methods for both storing and transferring data, ensuring that even if data is intercepted, it remains unreadable. Anonymizing the data by removing or encrypting identifying information helps protect user privacy further. Regular security audits and updates to protocols are essential in identifying and addressing vulnerabilities, thereby maintaining robust data security.

Another significant challenge is the potential for bias and unfairness in recommendation systems. These systems can sometimes reinforce existing biases, leading to unfair treatment and a lack of diversity in recommendations. For example, the system might start suggesting similar content to everyone in a demographic based on observed preferences, ignoring individual variations. To combat this, it's important to regularly check for biases by analyzing the model's performance across different user groups and ensuring it doesn't favor one group over another. Incorporating fairness-aware algorithms and using diverse, representative training data can improve fairness. Additionally, there's a risk of users becoming too dependent on recommendations, leading to a narrow and repetitive content experience. Designing the system to promote diverse content and encouraging user feedback can help mitigate this. By being transparent about how recommendations are made and promoting a variety of content, we can ensure a balanced and enriching user experience.

9 Conclusion

This project has demonstrated the significant potential of integrating the LLMs into recommendation systems to enhance personalization and user engagement. By converting user profiles and historical interactions into natural language prompts, we generated highly tailored and contextually rich recommendations with less user data compared to training on large datasets. Using the MovieLens 100k dataset, we achieved substantial improvements in precision, recall, and NDCG metrics compared to traditional models like Matrix Factorization, LightFM, and Item k-NN. The GPT-3.5-turbo model showed good movie recommendation and also explained why these movies were recommended, demonstrating its superior ability to provide relevant and accurate recommendations. This work advances the field of recommendation systems, offering a robust framework for developing more intuitive and user-centric personalization technologies which can be applied to cross domain recommendations.

References

[1] Youlin Xiang Yun Xiong Haofen Wang Jiawei Zhang Yunfan Gao, Tao Sheng. Chat-rec: Towards interactive and explainable llms-augmented recommender system. In *Information Retrieval (cs.IR); Computation and Language (cs.CL); Machine Learning (cs.LG)*, 2023.

[2] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

[3] Ziyan Jiang Eunah Cho Xiaojiang Huang Yanbin Lu Fan Yang, Zheng Chen. Palr: Personalization aware llms for recommendation. In *Information Retrieval (cs.IR); Artificial Intelligence (cs.AI); Computation and Language (cs.CL)*, 2023.

[4] Zhaopeng Qiu Hao Wang Hongchao Gu Tingjia Shen Chuan Qin Chen Zhu Hengshu Zhu Qi Liu Hui Xiong Enhong Chen Likang Wu, Zhi Zheng. A survey on large language models for recommendation. *Information Retrieval (cs.IR); Artificial Intelligence (cs.AI)*, 2023.

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

[6] Chang Zhou Jingren Zhou Hongxia Yang Zeyu Cui, Jianxin Ma. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv:2205.08084v2*, 2022.

[7] Zhengyi Yang Jiancan Wu Yancheng Yuan Xiang Wang Xiangnan He Jiayi Liao, Sihang Li. Llara: Large language-recommendation assistant. *Information Retrieval (cs.IR)*, 2022.

[8] Shayne Longpre Barret Zoph Yi Tay William Fedus Yunxuan Li Xuezhi Wang Mostafa Dehghani Siddhartha Brahma Albert Webson Shixiang Shane Gu Zhuyun Dai Mirac Suzgun Xinyun Chen Aakanksha Chowdhery Alex Castro-Ros Marie Pellat Kevin Robinson Dasha Valter Sharan Narang Gaurav Mishra Adams Yu Vincent Zhao Yanping Huang Andrew Dai Hongkun Yu Slav Petrov Ed H. Chi Jeff Dean Jacob Devlin Adam Roberts Denny Zhou Quoc V. Le Jason Wei Hyung Won Chung, Le Hou. Scaling instruction-finetuned language models. *Machine Learning (cs.LG); Computation and Language (cs.CL*, 2022.

[9] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. *Computation and Language (cs.CL)*, 2018.

[1] [2] [3] [4] [5] [6] [7] [8] [9]