Leveraging Large Language Models for Recommendation and Explanation

Italo Silva[1], Alan Said[2,*], Leandro Balby Marinho[1] and Martijn Willemsen[3]

[1]*Federal University of Campina Grande, Brazil*
[2]*University of Gothenburg, Sweden*
[3]*TU Eindhoven & JADS, The Netherlands*

Abstract

We demonstrate a proof of concept web application that recommends movies and generates explanations for the recommendations using Large Language Models (LLMs). Specifically, the application uses ChatGPT as both a movie recommender and model for generating explanations of the recommendations. The application gathers user preferences (liked movies), user goals (type of recommendations), and offers personalized recommendations and explanations tailored according to the used-defined goals. In this demo, we showcase how recommendations and explanations can be customized according to user-defined criteria, such as personalization, novelty, or popularity, and implemented using LLMs.

Keywords

recommender systems, explanations, large language models, recommendation goals

1. Introduction

In the past decades, recommender systems have become ubiquitous tools used in many connected scenarios, e.g., entertainment [1], professional work [2], and education [3], to name a few. More recently, Large Language Models (LLMs) such as GPT-3 and GPT-4 have witnessed a remarkable surge in adoption across diverse fields owing to their extraordinary natural language processing and generation capabilities. Initially engineered for tasks like text summarization, translation, and question-answering, LLMs have proved valuable beyond their initial realm. A notable application of LLMs is in the sphere of movie recommendations, where they not only suggest films but can also formulate insightful and tailored explanations [4, 5, 6]. The GPT-powered application ChatGPT has spiked tremendous interest from the research and practitioner communities in artificial intelligence, natural language processing, machine learning, as well as society at large. Used in the context of recommendation, ChatGPT can deliver a customized and personalized array of recommendations, bridging machine-generated suggestions and human-like interaction.

Woodstock'22: Symposium on the irreproducible science, June 07–11, 2022, Woodstock, NY
*Corresponding author.
italo.silva@ccc.ufcg.edu.br (I. Silva); alansaid@acm.org (A. Said); lbmarinho@dsc.ufcg.edu.br (L. B. Marinho); m.c.willemsen@tue.nl (M. Willemsen)
https://www.alansaid.com (A. Said); http://leandro.lsd.ufcg.edu.br/ (L. B. Marinho); https://martijnwillemsen.nl/ (M. Willemsen)
0000-0002-2929-0529 (A. Said); 0000-0001-7599-372X (L. B. Marinho); 0000-0001-5908-9511 (M. Willemsen)

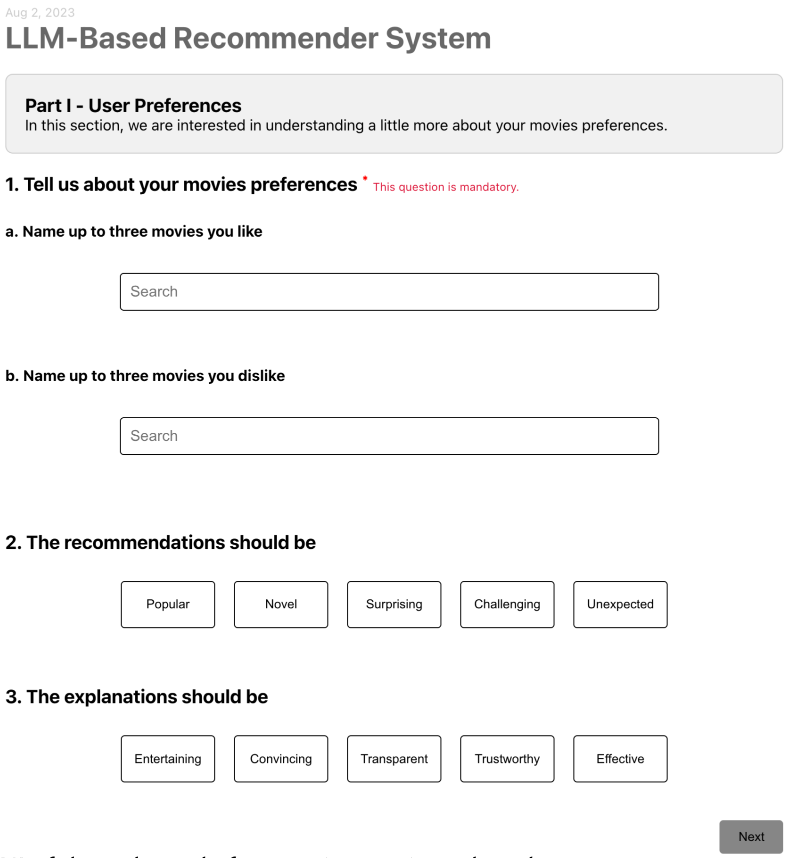


Figure 1: The UI of the web app before any interaction takes place

In addition to generating recommendations, LLM-powered applications like, e.g., ChatGPT, can be employed when developing interactive applications in a wide variety of domains. These applications can harness ChatGPT's conversational finesse to engage users, rendering the experience highly immersive, interactive, and personalized. For example, in e-commerce, ChatGPT can be deployed to recommend products based on customers' preferences and purchase history [6]. Likewise, in education, ChatGPT can deliver reading suggestions to users predicated on their skills, interests and reading habits. ChatGPT can also provide well-articulated explanations for its recommendations in each instance, helping establish trust between users and the application [7], and in doing so, increase the overall recommendation experience for the end users.

This proof of concept web application highlights the utilization of LLMs, particularly ChatGPT, in recommending movies and explaining the recommendations. ChatGPT is utilized to produce personalized recommendations on the one hand, and personalized explanations on the other hand, while also tailoring them based on explicit goals specified by the user.

The goals of this demo are to:

Goal 1. Showcase how LLMs like ChatGPT can be used in traditional recommendation scenarios.

Goal 2. Show how LLMs can easily extend traditional recommendation scenarios with their capabilities of following instructions.

Goal 3. Show how LLMs can generate personalized explanations of recommendations.

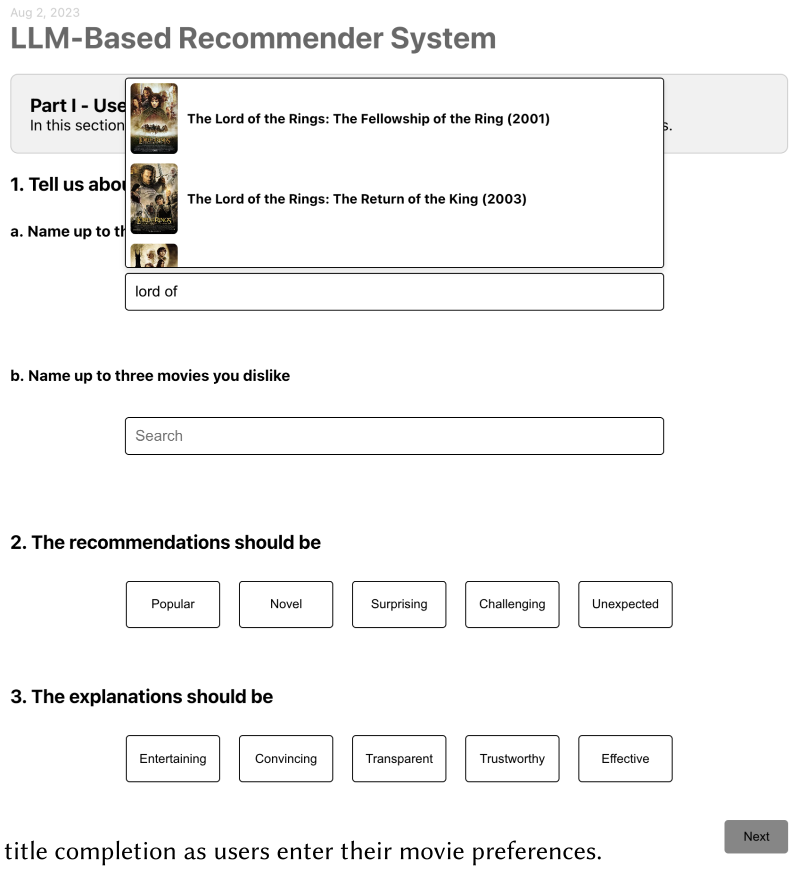


Figure 2: The title completion as users enter their movie preferences.

Goal 4. Demonstrate an intuitive user interface for recommendation and LLM applications.

As LLMs evolve, we can anticipate their deployment in increasingly inventive ways. Their proficiency in emulating human conversation and the ability to analyze massive amounts of data renders them invaluable assets in a rapidly expanding array of applications. Particularly in the context of movie recommendations, LLMs like ChatGPT could revolutionize recommendation by delivering content and curating an compelling and individualized experience for each end user. This exemplifies how LLMs, initially devised for generic language tasks, have become ideal tools for specialized and user-focused applications across increasingly various use cases, domains, and industries.

2. A Movie Recommender with User-defined Goals

We have developed a web application that leverages ChatGPT to generate movie recommendations and explanations of said recommendations based on user preferences. To craft recommendations, users are prompted to specify preferences regarding movies they like or dislike (refer to Fig. 1 and Fig. 2). Note that the application allows users to enter both liked and disliked movies in the preference elicitation stage. Having entered a set of movie preferences, users

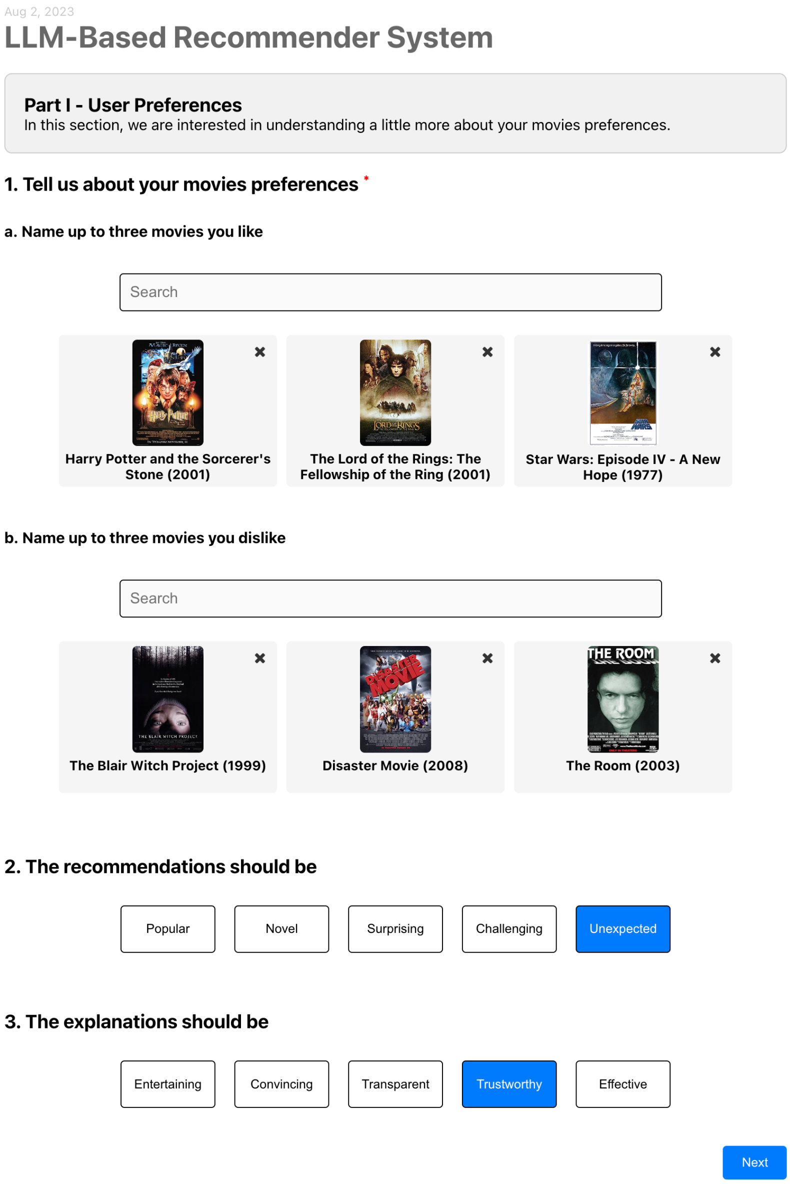
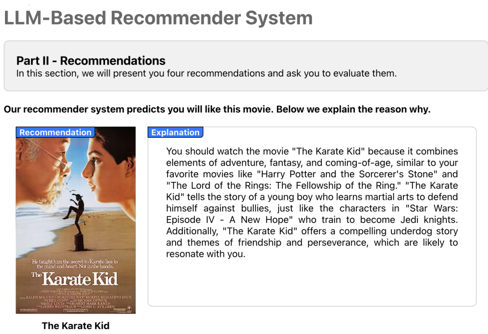
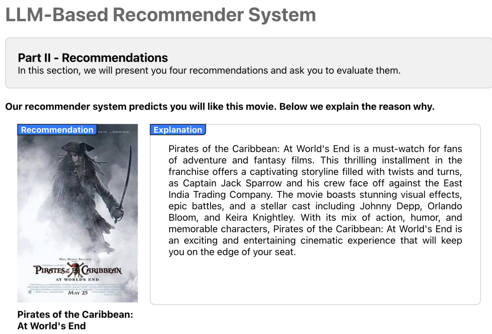


Figure 3: The webapp after a user has entered their movie preferences and their recommendation and explanation goals.

are also given the possibility to tailor their recommendations and explanations according to a set of goals (refer to Fig. 3), e.g., whether users seek surprises, popular movies, or novel content. While the goal alternatives for recommendations reflect commonly sought-for qualities in beyond-accuracy-based recommendation, the goals for the explanations, in turn, are inspired by the goals introduced by Tintarev and Masthoff [8].



(a) Recommendation with a personalized explanation.



(b) Recommendation with a non-personalized explanation.

Figure 4: Examples of a recommendation with a personalized explanation and a non-personalized explanation.

Table 1
User-based vs. Generic Explanations for a recommendation (*The Godfather*) to a user with the following preferences: **Likes**: "The Lord of the Rings: The Fellowship of the Ring", "Star Wars: Episode VI - Return of the Jedi", "Harry Potter and the Sorcerer's Stone". **Dislikes**: "Jaws", "The Exorcist", "The Blair Witch Project". The text where personalization takes place is red.

Personalized	**Generic**
The Godfather is a must-watch for someone who enjoys epic storytelling, complex characters, and gripping drama. This iconic film, directed by Francis Ford Coppola, shares similarities with your favorite movies like The Lord of the Rings, Star Wars, and Harry Potter. Just like those films, The Godfather immerses you in a richly detailed world, filled with memorable characters and a captivating narrative. It explores themes of power, loyalty, and family, much like the epic battles and quests in your favorite movies. The Godfather's masterful storytelling, brilliant performances, and unforgettable moments make it a timeless classic that you won't want to miss.	The Godfather is a must-watch movie for several reasons. Firstly, it is a cinematic masterpiece that has stood the test of time. The film is beautifully crafted, with a compelling storyline and exceptional performances by Marlon Brando and Al Pacino. Secondly, it offers a deep exploration of themes such as power, family, and loyalty, making it a thought-provoking experience. Lastly, The Godfather has had a significant influence on the film industry, shaping the crime genre and inspiring countless movies and TV shows. If you enjoy gripping storytelling, memorable characters, and a glimpse into the world of organized crime, The Godfather is a film you shouldn't miss.

3. Implementation of the Web Application

In this section we present the design considerations, implementation, and deployment of the web application.

Given a person with the following movie preferences.
Liked movies: **{liked movies}**
Disliked movies: **{disliked movies}**
Could you recommend four movies? Two of these movies should be a recommendation of a movie they must watch and two of movies they should avoid watching.

The recommendations must be **{rec goal}**.

Additional instructions:

1. Write the answer in the format of a JSON file with the attribute recommendations that is an array with four objects with the attributes: title, imdbID and shouldWatch.
2. The title attribute must be the title of the recommended movie and the shouldWatch must be a boolean indicating if it the movie is a recommendation of a movie the user should or not watch.
3. Do not recommend movies given in the user preferences answers.
4. Do not enumerate the movies.
5. Do not include any additional text besides the JSON.

Prompt 1: The prompt given to ChatGPT to receive recommendations. Note that the bold text in brackets is replaced with the preferences stated by the users.

3.1. Design and Considerations

While ChatGPT is a chatbot, this proof of concept has been implemented to resemble common media recommendation services, this entails, e.g., a preference elicitation step and a recommendation step, and (to some extent) a feedback step.

To support users in the preference elicitation phase, we integrated the OMDb API[1] to generate title-completion suggestions when a movie title is entered (Fig. 2). In addition to this, corresponding movie posters are displayed for visual verification as users begin typing movie titles.

Utilizing the preferences and goals given by the user, we formulate a prompt which is subsequently dispatched to the OpenAI Chat Completions API[2], powered by the GPT3.5-Turbo model. Each recommended movie is accompanied by an explanation based on the goal set by the user. Below the shown recommendation and expanation, the user can provide feedback on the quality of the recommendation and explanation.

The prompts formulated by our proof of concept follow the patterns shown in Prompt 1 for the recommendations, and Prompt 2 for the explanations. Note that the highlighted parts in the prompt examples correspond to variables that are replaced with the input provided by the users in the preference elicitation stage (see, Fig. 3).

[1]https://www.omdbapi.com/
[2]https://platform.openai.com/docs/guides/gpt/chat-completions-api

Given a person with the following movie preferences.
Liked movies: **`{liked movies}`**
Disliked movies: **`{disliked movies}`**
Why should someone with these preferences [not] watch the movie: **`{movie}`**?

The explanations must be **`{exp goal}`**.

Additional instructions:

1. Write the answer as a plain text with at least 300 and at most 350 characters and without any additional text besides the answer.
2. Write the explanation as if you was talking to someone, for example: 'You may like this and that'.

Prompt 2: The prompt given to ChatGPT to receive explanations of recommendations. Note that the bold text in brackets is replaced with the preferences stated by the users.

Fig. 4 displays examples of both a personalized explanation (Fig. 4a) and a generic one (Fig. 4b) based on the movies specified in Fig. 3. Table 1 shows both personalized and generic explanations for a recommendation received by a user. The personalized aspect of the explanation is highlighted. While the application does not intend to collect an entire catalog of user preferences, even the relatively few collected user preferences are visible in the personalized explanation, with ChatGPT reasoning about movie qualities which would likely require more user preferences in order to be learned by an recommender not using LLMs for recommendation.

3.2. Implementation

The web application is built using Python, with Flask as the web framework. The backend service compiles prompts by processing input from the user and then sends an API call to ChatGPT (see, e.g., Prompt 1 and 2). This introduces a slight delay of about one minute before the recommendations and explanations are presented to the user.

The design of the prompts has been inspired by Gao et al. [9] and Wang and Lim [5]. The application is deployed in an AWS instance accessible to anyone[3].

4. Future Work

While this proof of concept web application is a simple demonstration of how LLMs can be leveraged for recommendation and explanation purposes, the usage of LLMs in this context remains novel, with much work left ahead. In the context of this demo, future works include further analysis of the design and implications of goal statements on both recommendations and explanations. This demo serves as a seed for future research integrating LLMs into the

[3]A link will be made available during the workshop allowing attendees to try the application on their own devices.

recommendation process. With the rapid adoption of LLMs as a *tool for everything*, the potential research directions in this space are plentiful. Our goal is to provide a simple and palpable example of how the capacities of LLMs can be used and leveraged to power the new generation of recommender systems.

References

[1] C. A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Trans. Manage. Inf. Syst. 6 (2016). URL: https://doi-org.ezproxy.ub.gu.se/10.1145/2843948. doi:10.1145/2843948.

[2] F. Abel, We know where you should work next summer: Job recommendations, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 230. URL: https://doi-org.ezproxy.ub.gu.se/10.1145/2792838.2799496. doi:10.1145/2792838.2799496.

[3] H. Drachsler, K. Verbert, O. C. Santos, N. Manouselis, Panorama of Recommender Systems to Support Learning, Springer US, Boston, MA, 2015, pp. 421–451. URL: https://doi.org/10.1007/978-1-4899-7637-6_12. doi:10.1007/978-1-4899-7637-6_12.

[4] Y. Zhang, H. DING, Z. Shui, Y. Ma, J. Zou, A. Deoras, H. Wang, Language models as recommender systems: Evaluations and limitations, in: I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop, 2021.

[5] L. Wang, E.-P. Lim, Zero-shot next-item recommendation using large pretrained language models, 2023. arXiv:2304.03153.

[6] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, Y. Zhang, Is ChatGPT a Good Recommender? A Preliminary Study, 2023. URL: http://arxiv.org/abs/2304.10149, arXiv:2304.10149 [cs].

[7] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, J. Xu, Uncovering ChatGPT's Capabilities in Recommender Systems, 2023. URL: http://arxiv.org/abs/2305.02182. doi:10.48550/arXiv.2305.02182, arXiv:2305.02182 [cs].

[8] N. Tintarev, J. Masthoff, A survey of explanations in recommender systems, in: 2007 IEEE 23rd International Conference on Data Engineering Workshop, 2007, pp. 801–810. doi:10.1109/ICDEW.2007.4401070.

[9] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, J. Zhang, Chat-rec: Towards interactive and explainable llms-augmented recommender system, 2023. arXiv:2303.14524.