Potentials of Combining Local Knowledge and LLMs for Recommender Systems

Thomas E. Kolb[1,*,†], Ahmadou Wagne[1,†], Mete Sertkan[1] and Julia Neidhardt[1]

[1]*Christian Doppler Laboratory for Recommender Systems, TU Wien, Vienna, Austria*

Abstract

LLMs have revolutionized the understanding and generation of natural language, offering new possibilities for enhancing recommendation systems. In previous studies, LLMs exploit their global knowledge to provide zero- or few-shot recommendations. In this work, we aim to highlight the opportunities that LLMs pose to enrich the field of recommender systems combined with local knowledge. We propose to view recommender systems combined with LLMs from a broader perspective, recognizing them not merely as another method to replace existing recommendation approaches, but rather as a complementary and powerful approach to enhance and augment the overall recommendation process.

Keywords

recommender systems, knowledge-aware recommendations, LLMs

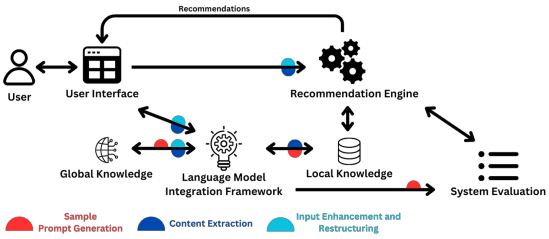

Figure 1: Recommender system architecture that combines global and local knowledge

1. Introduction

The advent of generative large language models (LLMs) like Generative Pre-trained Transformer (GPT)[1], LLaMA [1] or PaLM[2] introduced new possibilities to understand and generate natural language that can support the recommendation process at various stages. Early studies that investigated the application of e.g. ChatGPT[3] for recommender systems mostly focused on using it directly as a zero-shot or few-shot recommender system or on exploiting the language understanding and generation capabilities to provide explainable recommendations [2, 3, 4]. This paper provides insights into possible ways to use LLMs that differ from approaches that only rely on the global knowledge of pre-trained models and directly recommend items based on that. This is often not applicable in a real-world scenario, where you have a fixed set of items to choose from and want to ensure factual recommendations. Moreover, our goal is to provide textual metadata about a dedicated set of items, which we call local knowledge, for our recommendations, but use the global knowledge of LLMs to support other steps of the pipeline of a system. To further investigate this, we conducted a pilot study involving students of a Masters course that applied this approach practically. This method can furthermore overcome problems when the global knowledge does not hold any information about an item in one's own stock. Figure 1 highlights these different components and emphasizes that modern recommender systems consist of a whole system architecture of which the recommendation engine is only one part. In this work, local knowledge is defined, as the data employed exclusively by the language model integration framework or the recommendation engine, independent of any access to the global knowledge of the LLM; such knowledge can be stored through diverse methods (e.g. vector stores, text files, etc.). This local knowledge is used as the basis to retrieve items for recommendations based on e.g. a similarity search. LLMs in general additionally often face the issue of hallucination [5, 6], which is especially undesirable in this context, because you want to recommend items that are actually available. Retrieval-Augmented Generation [7] has already been shown to alleviate these problems in the domain of information retrieval to provide more factual and knowledge-based results.

5th Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) @ RecSys 2023, 18th-22nd September 2023, Singapore
*Corresponding author.
†These authors contributed equally.
thomas.kolb@tuwien.ac.at (T. E. Kolb); ahmadou.wagne@tuwien.ac.at (A. Wagne); mete.sertkan@tuwien.ac.at (M. Sertkan); julia.neidhardt@tuwien.ac.at (J. Neidhardt)
0000-0002-2340-0854 (T. E. Kolb); 0009-0009-9314-206X (A. Wagne); 0000-0003-0984-5221 (M. Sertkan); 0000-0001-7184-1841 (J. Neidhardt)

[1]https://openai.com/research/gpt-4
[2]https://ai.google/discover/palm2/
[3]https://chat.openai.com/

2. Opportunities and Challenges

We anticipated a lot of potential in using LLMs in various stages of recommender systems to support finding suitable recommendations from local knowledge. This section highlights the potentials that should be further explored in our pilot study and reflect on challenges that come with them.

2.1. Input Enhancement and Restructuring

When creating recommendations for a user, we can process different types of signals as input for a system [8, 9]. Those can be user profiles, metadata of relevant items or direct input from a user. In cases where the user can input queries, they often provide simplistic statements with sparse textual information. To address this limitation, we see potential to automatically enrich those with additional information, by using generative models. The practice of query expansion has shown potential in various natural-language-processing tasks [10, 11]. One approach involves using knowledge from pre-trained LLMs to augment user input and provide richer information about the underlying intent and alleviate cold start problems. This can be used to generate user profiles in textual form that are subsequently used to query the local knowledge to retrieve similar or fitting items to the enhanced profile. In a conversational scenario, this can be supported by the means of the LLM to provide the user with dynamic questions in real-time to get more information about their intent directly from them.

Challenges arise when the input structure differs from the meta-information about target items. An example setting would be recommending job listings based on resumes. Job listings can follow a structure with different fields that hold information about tasks etc. In such cases, LLMs offer promising potential to restructure resumes and create synthetic job listings or vice versa. Additionally, LLMs can help imputing missing values within structured content, such as generating a short pitch or personal description from working experiences and interests in resumes. Experimental investigations in this direction can include prompt engineering to align the enhancement or restructuring process across multiple user sessions, given the non-deterministic nature of generated responses.

2.2. Content Extraction and Sample Prompt Generation

Another notable application revolves around the generation of supplementary content based on target items, which serves various purposes. An example is the usage of LLMs to extract topics or summaries from target items. In the context of news, LLMs have shown to be on level with human-created summaries [12]. Automatically generating summaries or topics holds benefits both in terms of efficiency and effectivity. In a large local data base of news articles, processing quickly becomes expensive, when assessing the similarity between a user profile's content or a natural language user query and the full-text articles on every invocation. This process can be alternatively executed on a concise summary, containing significantly fewer tokens while incurring only minimal information loss. Generating topics holds the benefit of providing further information that can be exploited for recommendations that better fit the user's needs. Topics can include reading motives of users for news or books or occasions for outfits in a fashion recommendation scenario. The extracted topics can then either be inferred from user input or directly asked for by the system to then perform a directed search on the local knowledge.

A further application scenario of LLMs based on the information extraction aids the process of evaluating recommender systems. To reduce the need of real world data, which might not be available, if a system is not launched yet or annotated historical data, LLMs can reverse engineer sample prompts based on items to check the consistency of the recommendation engine. This is not limited to user prompts, but there is furthermore the possibility to create synthetic user profiles for example. Re-feeding such synthetic samples to the system hold potential to provide new means for evaluation. The usage and nature of prompts by real users will most likely differ from those synthetic prompts and ways to create suitable samples have to be further researched.

2.3. Further Use Cases

We want to mention additional use cases that tackle open problems in the recommender systems field or can be combined with traditional approaches for future endeavours. When using LLMs to hold conversations with users, we can use the capabilities of LLMs to extract the intent of users and build profiles dynamically where the system generates questions in real time that aim at acquiring specific missing information or infers it from a conversation flow. The conversational setting furthermore enables the incorporation of real time feedback and iterative adaption and refinement of recommendations. This process involves presenting recommendations based on prompts or the chat history and allowing users to provide ratings or rankings for the suggested items. The system leverages this feedback as additional information about user preferences, subsequently enhancing the precision of following recommendations, resulting in more personalised and contextually relevant recommendations.

A major challenge for all mentioned scenarios is that one has to engineer a set of prompts or prompt templates that make sure that the model executes the concerning task as intended and delivers responses consistently in the right format to be further processed and stable in

a deployed system. However the incorporation of local knowledge mitigates problems that arise with the reliance on global knowledge, like bias or hallucination of LLMs, which is also seen as a general concern as highlighted by the Digital Humanism Initiative[4]. In a recommendation scenario we want to make sure that items are available and metadata about them is correct and reliable.

3. Pilot Study

To investigate the raised challenges and opportunities in Section 2 we conducted a pilot study[5] in one of our lectures. The group tasks were structured in a way to encourage the students to use innovative and new approaches to solve the given task. Each group was tasked with leveraging the capabilities of a LLM, in this pilot study, the OpenAI GPT API[6] in combination with a popular language model integration framework[7] for utilizing LLMs for different downstream tasks e.g. the usage of local knowledge. The research potential of the opportunities and challenges previously addressed is supported by the findings of the pilot study conducted.

New insights into query enhancement and restructuring were gained through four different projects: (1) matching a set of resumes with a dataset of job listings, (2) creating a system, which is able to recommend companies based on certain preferences (i.e. to understand competitive suppliers in the marketplace), (3) recommending clothes to wear on a particular occasion, and (4) building a recommender system to recommend books based on a given book. Project (1) utilized LLMs to *generate structured JSON Files from resumes* and to complete null fields based on other content. They used a creative approach to *generate job listings from resumes* to match them with the jobs within the job listing. In addition *query extension was utilized to enhance or complement the resumes* by accessing the global knowledge of the LLM. In task (2) the assumption was that a user would use a rather simplistic prompt to search for competing companies. They *enhanced this query* by prompting GPT for an enhanced version of the given query. The approach was to provide GPT with the original prompt, including a keyword list of the user preferences based on the behavior within the session. Subsequently, the user is provided with multiple recommendations for generated search prompts from which they can choose the one that best reflects their intent. The selected prompt is then used for the further recommendation process. Project (3) solved the given problem by providing the user with a chat interface. The user can chat with the system and submit their preferences. The system internally *rewrites the chat history as a single question* and feeds it back to GPT. This has the goal to refine the fashion article recommendations provided by the system. In their internal evaluation they state that they were rather satisfied with the recommended items, which resulted in a time saving and the system showed them clothing options which they would not have considered otherwise.

Content extraction and sample query generation were tackled by two different approaches: (1) recommending news items to a certain topic based on a news data-set, and (2) recommending news items based on a certain reading motive or category. In task (1) GPT was used for *topic extraction* which was applied on the user prompts and news articles. They propose a personalised recommender system which stores the extracted topics from user queries and articles in a joint vector store. In addition, random noise following a normal distribution is introduced to diversify recommendations. With more user queries the user feedback is weighted more than the noise, going by the assumption that the system gradually understands its users better. Task (2) was very similar, but this time the goal was to identify reading motives. The proposed solution was a combination of embedding all articles with the help of Ada[8] and *extracting reading motives from articles with the help of GPT* to match them with user preferences.

4. Conclusion

In conclusion, our work highlights the abundant potential of leveraging local knowledge in conjunction with LLMs to advance recommender systems. We propose the need for a comprehensive exploration of this potential, transcending the early trends of using LLMs to replace traditional recommenders or focusing on explainable recommendations. Recognising modern recommender systems as multifaceted ecosystems, where the recommendation algorithm represents only one component, we propose investigating mentioned aspects such as input enhancement and restructuring, content extraction, and sample prompt generation in future research endeavors. By doing so, we can benefit from the ever-evolving capabilities of LLMs, while also mitigating their drawbacks through the incorporation of trusted local knowledge. This approach promises to uncover new advancements in personalised and knowledge-aware recommendations, enriching the overall user experience and amplifying the benefits of using LLMs in recommender systems.

Acknowledgments

This research is supported by the Christian Doppler Research Association (CDG).

[4] https://caiml.dbai.tuwien.ac.at/dighum/statement-of-the-digital-humanism-initiative-on-chatgpt/

[5] The pilot study was designed as an optional track within the lecture to comply with the responsible research practices at TU Wien.

[6] https://openai.com/blog/openai-api

[7] https://python.langchain.com/

[8] https://openai.com/blog/new-and-improved-embedding-model

References

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models (2023).

[2] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, J. Zhang, Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System (2023).

[3] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, Y. Zhang, Is ChatGPT a Good Recommender? A Preliminary Study (2023).

[4] Z. Cui, J. Ma, C. Zhou, J. Zhou, H. Yang, M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems (2022).

[5] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, P. Fung, A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity (2023).

[6] G. Marcus, The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence (2020).

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2020).

[8] F. Ricci, L. Rokach, B. Shapira, Introduction to Recommender Systems Handbook, in: Recommender Systems Handbook, Springer US, Boston, MA, 2011, pp. 1–35. doi:10.1007/978-0-387-85820-3{_}1.

[9] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems 46 (2013) 109–132. doi:10.1016/j.knosys.2013.03.012.

[10] R. Jagerman, H. Zhuang, Z. Qin, X. Wang, M. Bendersky, Query Expansion by Prompting Large Language Models (2023).

[11] L. Wang, N. Yang, F. Wei, Query2doc: Query Expansion with Large Language Models (2023).

[12] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, T. B. Hashimoto, Benchmarking Large Language Models for News Summarization (2023).