Beyond Words: Can ChatGPT support state-of-the-art Recommender Systems?*

Dario Di Palma[1,*], Giovanni Servedio[1,*], Vito Walter Anelli[1], Giovanni Maria Biancofiore[1], Fedelucio Narducci[1], Leonarda Carnimeo[1] and Tommaso Di Noia[1]

[1]*Politecnico di Bari*

Abstract

Large Language Models (LLMs) have recently demonstrated impressive capabilities across a range of natural language processing tasks. Notably, ChatGPT has exhibited superior performance in numerous tasks, particularly under zero and few-shot prompting conditions. Motivated by these successes, the Recommender Systems (RS) and Information Retrieval research communities have started exploring the potential applications of ChatGPT in recommendation and information filtering scenarios.

This study investigates the performance of ChatGPT-3.5 and ChatGPT-4 in recommendation tasks under zero-shot conditions, employing a role-playing prompt. We specifically analyze the models' ability to re-rank recommendations in three domains: movies, music, and books. Our experiments indicate that ChatGPT excels in re-ranking tasks, providing high-quality recommendations. Furthermore, we measure the similarity between ChatGPT's recommendations and those generated by other recommendation systems, offering insights into how ChatGPT can be positioned within the RS landscape.

Keywords

ChatGPT, Recommender Systems, Evaluation, Re-ranking

1. Introduction

The rapid increase in human-generated data, particularly unstructured text, has significantly transformed the digital world. The World Wide Web now serves as a vast repository of diverse information, representing both individual and corporate needs, opinions, and knowledge. However, extracting value from this immense resource remains a major challenge.

Addressing this challenge involves the use of Natural Language Processing (NLP) techniques. NLP encompasses a range of computational methods [1] designed to analyze, interpret, and derive meaning from natural language. By employing these techniques, professionals across fields like data science, IT, marketing, and social research can tap into the potential of this extensive unstructured data.

The ease of sharing and accessing information has significantly increased the speed of information dissemination, but it also brings concerns such as information overload (e.g. [2]), and issues related to quality, privacy, and security.

IIR2024: 14th Italian Information Retrieval Workshop, 5th - 6th September 2024, Udine, Italy

*Corresponding authors.

dario.dipalma@poliba.it (D. D. Palma); giovanni.servedio@poliba.it (G. Servedio); vitowalter.anelli@poliba.it (V. W. Anelli); giovannimaria.biancofiore@poliba.it (G. M. Biancofiore); fedelucio.narducci@poliba.it (F. Narducci); leonarda.carnimeo@poliba.it (L. Carnimeo); tommaso.dinoia@poliba.it (T. D. Noia)

Various methods, including Information Retrieval (IR) [3] and Recommender Systems (RSs) [4], are designed to sift through large datasets to find valuable information. However, their success largely depends on efficiently filtering out irrelevant or low-quality content to ensure that users can quickly find reliable and pertinent information.

IR systems have integrated advanced NLP techniques like semantic analysis and natural language understanding to provide contextually relevant information beyond simple keyword matching, allowing for a deeper understanding of user queries.

Recommender Systems have also advanced in predicting user preferences and delivering personalized content recommendations [5]. By analyzing past behaviors, interactions, and preferences, RSs curate content that aligns with individual tastes. Incorporating NLP techniques in RSs enhances their ability to capture user preferences with greater detail, resulting in highly relevant recommendations.

Additionally, recent research [6, 7, 8] underscores the benefits of interactive systems in improving the quality and relevance of information and enhancing user satisfaction and experience [9]. These systems, through human-like dialogues, can retrieve more precise data tailored to individual preferences, leading to a more personalized user experience.

This shift towards conversational interfaces has fueled the rise of digital assistants like Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri [10]. Their ability to understand and process natural language queries in real-time has revolutionized user interaction with technology, making information access more intuitive and efficient.

The development of advanced Language Models, particularly Large Language Models (LLMs) like Generative Pre-trained Transformer 3 (GPT-3) [11], has greatly enhanced interaction with digital systems [12], enabling unprecedented access to information through natural language queries.

These models represent a new era in machine comprehension and language generation, allowing for conversations with remarkable naturalness and depth. Their capabilities go beyond simple query processing, enabling dynamic and nuanced dialogues similar to human interactions.

While GPT-3 has shown significant advancements in generating human-like text, the introduction of ChatGPT on November 30, 2022[1], marked a milestone with an AI model capable of engaging in dialogue like never before, covering a wide array of tasks [11]. Researchers are keenly exploring ChatGPT's potential across various applications, particularly in recommendation tasks [13, 14, 15, 16].

While there is increasing interest in integrating ChatGPT with recommender systems to improve accuracy [17, 18] or fairness [19, 20], some areas, such as the system's ability to re-rank item lists, remain under-explored.

We investigate the performance of ChatGPT-3.5 and ChatGPT-4 in recommendation tasks under a zero-shot role-playing prompt condition. This approach allows us to evaluate these models' effectiveness as re-rankers in three different domains: Books, Movies, and Music (using datasets like Facebook Books [21], MovieLens [22], and Last.FM [23]).

Moreover, since ChatGPT is a Large Language Model and not a dedicated Recommender System, we explore its ability to suggest items by comparing its recommendation lists with those generated by content-based, collaborative filtering, and hybrid recommenders. This

[1] https://openai.com/blog/chatgpt/

analysis aims to determine whether ChatGPT favors content-based or collaborative filtering methods in its recommendation generation. The goal is to gain insights into the approach ChatGPT mimics during the recommendation process.

Our contributions can be summarized as follows:

- We evaluate the performance of ChatGPT-3.5 and ChatGPT-4 on re-ranking a list of recommendations across three domains: Books, Movies, and Music.
- We investigate the underlying methodology employed by ChatGPT in generating recommendations, aiming to understand whether the model demonstrates an inclination towards content-based, collaborative, or hybrid recommenders to gain insights into its recommendation process.

2. Related Work

Integrating Large Language Models (LLMs) into recommender systems has garnered significant attention due to their powerful generative capabilities. However, much of the research in this area remains preliminary. Notable examples include M6-Rec [24], which leverages the M6 LM [25] for various recommendation tasks by framing the recommendation process as a language understanding or generation task. Similarly, P5 [26] uses personalized prompt templates to integrate user-item information, enabling predictions in a zero-shot or few-shot manner. These approaches, however, create ad-hoc solutions rather than directly utilizing generic LLMs like ChatGPT.

Wu et al. [27] employ LMs (e.g., BERT, GPT-2) and LLMs (e.g., T5, LLaMA) to address the cold start issue, highlighting the critical role of prompt design. Building on this concept, our research focuses on re-ranking items rather than recommending them individually. Zhang et al. [28] fine-tune T5 for sequential recommendations, demonstrating comparable ranking abilities to zero-shot methods like GPT-3.5.

He et al. [29] conduct an empirical study on conversational recommendations using GPT-3.5, GPT-4, and LLAMA, finding that LLMs outperform fine-tuned Conversational Recommender System (CRS) models. Kang et al. [30] show that zero-shot LLMs can be cost-effective compared to fine-tuned models, prompting further exploration of ChatGPT's ranking capabilities.

Early work by Zhang et al. [31] used GPT-2 for session-based recommendations, showing that pre-trained LM-based methods can perform well in zero-shot settings. Recent studies, such as GPT4Rec by Li et al. [32] and Wang and Lim [33], indicate that LLMs can effectively handle recommendation tasks through strategic prompting.

ChatREC by Gao et al. [13] introduced a ChatGPT-augmented recommender system using interactive conversations to generate Top-N recommendations.

Hou et al. [16] explored ranking tasks with LLMs in zero-shot settings, identifying position bias issues in prompts. Sanner et al. [34] assessed LLM-based RS abilities for Top-N recommendations, highlighting the effectiveness of zero-shot settings in cold-start scenarios. Li et al. [35] developed BookGPT for book recommendations, using Role-Play prompting to achieve

accuracy comparable to baselines, although their focus was limited. Dai et al. [17] examined ChatGPT's performance in various recommendation tasks, showing balanced performance in non-rating tasks. Xu et al. [36] confirmed the effectiveness of Role-Playing prompting in zero-shot scenarios for LLM-based RSs, focusing on accuracy.

Our work is the first to investigate re-ranking capabilities, examining ChatGPT's ability to generate lists similar to content-based and collaborative filtering methods.

3. Experiments

The following sections detail the methodology and experimental setup used in our research. Specifically, Section 3.1 reviews the datasets employed and the necessary pre-processing steps to comply with ChatGPT's constraints. Section 3.2 presents the models used for similarity checks and the experimental settings. Finally, Sections 3.3 and 3.4 discuss the results from evaluating ChatGPT as a re-ranker and its similarity to other recommenders. Each experiment is designed to address the following research questions:

RQ1. Can ChatGPT effectively re-rank and enhance recommendations by utilizing user history? (Explored in Section 3.3)

RQ2. How closely do the recommendation lists generated by ChatGPT match those produced by Collaborative Filtering and Content-based Recommender Systems? (Examined in Section 3.4)

3.1. Pre-processing Phase

In the domain of Recommender Systems (RSs), the most common method to evaluate a model's performance is through offline experiments using existing datasets, typically historical data or logs. For our study, we utilized three well-known and publicly accessible datasets from different domains: books, music, and movies.

Specifically, we used the Facebook Books [21], Last.FM [23], and MovieLens [22] datasets. Despite their popularity in literature, we had to consider the token limits of the ChatGPT API. To accommodate this limitation, we applied specific preprocessing steps to adjust the user interaction histories to the context length imposed by the API.

Consequently, we adopted an iterative-10-core strategy on the users and items within the datasets, retaining only those with at least ten occurrences. However, upon analyzing each dataset, we found that some users' interaction histories in MovieLens exceeded the maximum context length. Therefore, additional preprocessing was necessary to reduce the number of interactions. We set a threshold of 200 interactions to address the context constraints, resulting in a modified MovieLens dataset.

Table 1 presents the statistics of these datasets before and after preprocessing.

Here is a brief overview of each dataset:

Facebook Books Dataset. The Facebook Books dataset[2] was released for the Linked Open

[2]https://github.com/sisinflab/LinkedDatasets/

Table 1
Dataset statistics before and after pre-processing with max context length filtering and $k - core \geq 10$.

Dataset	Interaction	Users	Items	Sparsity	Interaction	Users	Items	Sparsity	Content	
	before pre-processing				*after pre-processing*				*type*	*features*
MovieLens	100836	610	9724	98.30%	42456	603	1862	96.22%	genre	20
Last.FM	86608	1892	12133	99.62%	49171	1797	1507	98.18%	genre	9748
FB Books	18978	1398	2234	99.74%	13117	1398	2234	99.58%	genre, author	1970

Data Challenge 2015[3] and covers the book domain. It includes implicit feedback and item-feature mappings to DBpedia for each book, allowing the retrieval of data content such as book genres and relevant author information.

Last.FM Dataset. The Last.FM dataset contains user-artist play data from the Last.FM online music system, released during the HETRec2011 Workshop[4] [23]. It includes information on social networking, tagging, artists, and music listening habits of 2000 users.

MovieLens Dataset. The MovieLens dataset is widely used in the Recommender Systems community [22]. Various versions are available[5], but our study utilized the MovieLens 100k version, which contains movie ratings on a 1-5 scale.

3.2. Experimental Setting

To conduct the experiments, we compared the performance of ChatGPT-3.5 and ChatGPT-4 models against state-of-the-art recommender systems. To ensure a fair comparison, we conducted extensive Bayesian hyper-parameter optimization for each baseline to determine the optimal configuration. For complete reproducibility, we utilized the Elliot recommendation framework [37].

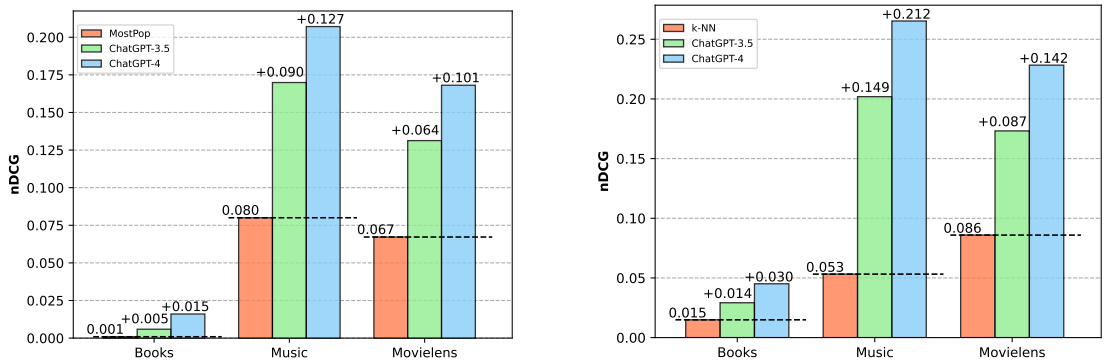
The models used to evaluate ChatGPT against state-of-the-art recommenders are categorized into Collaborative Filtering, Content-Based Filtering, and Non-personalized approaches:

- **Collaborative Filtering**: EASE^R [38], RP$^3_\beta$ [39], ItemKNN [40], UserKNN [41], LightGCN [42], MF2020 [43], NeuMF [44].
- **Content-Based Filtering**: VSM [45], AttributeItemKNN [46], AttributeUserKNN [46].
- **Non-personalized** Random Model, MostPop Model.

The evaluation employs the all unrated items protocol [47, 48], where the set of recommendable items for each user includes all items except those previously interacted with by the user. The datasets are divided into training and test sets, with 80% of the user-item interactions used for training and 20% reserved for testing.

While the recommendation baselines can be implemented using the open-source Elliot framework, generating recommendations from the ChatGPT models requires using the OpenAI API. Specifically, for each user in each dataset, we craft a prompt to obtain a list of recommended items.

[3] https://2015.eswc-conferences.org/program/semwebeval.html
[4] https://grouplens.org/datasets/hetrec-2011/
[5] https://grouplens.org/datasets/movielens/



(a) Re-ranking the Most Popular items.

(b) Re-ranking Nearest Neighbors items.

Figure 1: Grouped bar charts illustrating the performance of the ChatGPT models across three domains. Each grouped bar chart displays the baseline nDCG score and the improvement in terms of nDCG score. Higher improvements in nDCG indicate a better ability of the model to re-rank and personalize a list of items according to individual user preferences.

3.3. Re-ranking Evaluation

This section investigates ChatGPT's potential to re-rank a list of recommendations, i.e., the ability of a recommender to improve the quality of recommendations by adjusting the order in which items are presented to the user [49].

We defined two experimental conditions to assess ChatGPT's re-ranking capability: (a) asking ChatGPT to re-rank the most popular items in the datasets based on the user's history, and (b) asking ChatGPT to re-rank a pre-filtered list of suitable items for each user using a k-Nearest Neighbors (k-NN) algorithm.

Re-ranking Prompt

Given a user, act as a Recommender System. You know that the user likes the following items ordered by preference: {history of the user}. Re-rank this list of items into a top-50 recommendations: {list of items to re-rank}

To conduct the experiments, we used a Role-Play Prompting approach [50] to query ChatGPT. For condition (a), re-ranking the most popular items, we used the prompt shown above, ordering the items from most to least popular. For condition (b), re-ranking the nearest neighbor recommendations, we used the same prompt, but the item order was determined by the k-NN algorithm.

Building on these experimental conditions, this section aims to address the following research question:

RQ1. Can ChatGPT effectively re-rank and enhance recommendations by utilizing user history?

Figures 1a and 1b illustrate the performance of ChatGPT-3.5 and ChatGPT-4 models across three domains: Books, Music, and Movies, compared to the Most Popular and k-NN methods. Performance is evaluated using nDCG, a measure of ranking quality that accounts for both the relevance and the position of items in the ranked list. Higher nDCG values signify better performance in ranking relevant items higher.

From the bar graphs, we can draw the following observations:

- Across all three domains, ChatGPT-4 consistently achieves higher nDCG scores than ChatGPT-3.5 and the baseline methods, regardless of the experimental conditions. This indicates that ChatGPT-4 is more effective at ranking items according to users' preferences.
- The type of setting also influences the results. For the Most Popular items, shown in Figure 1a, the MostPop baseline exhibits a lower nDCG due to insufficient user influence. However, initial filtering using k-NN results in higher nDCG scores, indicating a step towards personalization. Despite this, the ChatGPT models, particularly ChatGPT-4, further enhance recommendation performance by better understanding user preferences.

In summary, to address RQ1, we analyze the nDCG values and observe that ChatGPT's re-ranking improves recommendation performance, providing a positive answer. While the study does not delve into the reasons behind this improvement, we hypothesize that it arises from GPT's extensive knowledge base and its ability to understand user preferences and the relevance of items. Future research will focus on exploring the explainability of the deep learning model behind ChatGPT and its efficacy in re-ranking recommendations. However, a detailed explainability analysis is beyond the current study's scope.

3.4. Recommendation List Similarity

This section examines the similarity between the recommendation lists generated by ChatGPT and those produced by Content-based or Collaborative Filtering Recommender Systems (RSs) using a role-play prompt defined below.

Top-N Recommendation Prompt

Given a user, as a Recommender System, please provide only the names of the top 50 recommendations. You know that the user likes the following items: {history of the user}

Our hypothesis is that ChatGPT will leverage content features of the items, but we also suppose it will learn collaborative information during the training process. To validate this hypothesis, we compare the similarity between these recommendation lists. Specifically, we aim to answer the following research question:

RQ2. How closely do the recommendation lists generated by ChatGPT match those produced by Collaborative Filtering and Content-based Recommender Systems?

To address this question, we use two types of metrics: the Jaccard Index [51], which measures the size of the intersection between two sets of recommended items for each user, disregarding the order of items, and Rank-biased Overlap (RBO) [52], which measures the similarity between two ranked lists, considering the order of items.

The results, presented in Tables 2, 3, and 4, are divided by domain. For each domain and model, the average metrics are calculated on a per-user basis, highlighting the types of recommender systems, namely Collaborative Filtering (CF) and Content-Based Filtering (CBF) methods.

For the **Facebook Books** dataset, ChatGPT-4's recommendations show a high degree of similarity with those of ChatGPT-3.5, followed by $\text{EASE}^R_{(CF)}$, $\text{LightGCN}_{(CF)}$, and MostPop,

Table 2
Comparative analysis of **Lists Similarity** between ChatGPT-3.5 and ChatGPT-4 with the baselines on **Facebook Books**. Arrows indicate whether higher (↑) or lower (↓) values are desirable for each metric. Best values are in bold. The second-best values are underlined. The baselines are statistically significant based on paired t-tests ($p < 0.05$) except for the values denoted with *(for ChatGPT-3.5) and †(for ChatGPT-4).

ChatGPT-4			ChatGPT-3.5		
Model	**Jaccard Index ↑**	**RBO ↑**	**Model**	**Jaccard Index ↑**	**RBO ↑**
Random†	0.0094	0.0092	Random*	0.0079	0.0073
NeuMF	0.0364	0.0467	NeuMF	0.0280	0.0360
RP^3_β	0.0576	0.1463	RP^3_β	0.0386	0.1018
AttributeItemKNN	0.0643	0.1236	AttributeItemKNN	0.0390	0.0631
ItemKNN	0.0647	0.1101	VSM	0.0446	0.0695
VSM	0.0747	0.1383	ItemKNN	0.0448	0.0699
UserKNN	0.0885	0.1424	UserKNN	0.0684	0.1169
MF2020	0.1214	0.1468	MF2020	0.0914	0.1112
AttributeUserKNN	0.1242	0.2081	AttributeUserKNN	0.0950	0.1622
MostPop	0.1410	0.1904	MostPop	0.1231	0.1898
LightGCN	0.1441	0.1938	LightGCN	0.1247	0.1911
$EASE^R$	0.1637	0.2344	$EASE^R$	0.1351	0.2142
ChatGPT-3.5	**0.2274**	**0.3637**	ChatGPT-4	**0.2274**	**0.3637**

Table 3
Comparative analysis of **Lists Similarity** between ChatGPT-3.5 and ChatGPT-4 with the baselines on **Last.FM**. Arrows indicate whether higher (↑) or lower (↓) values are desirable for each metric. Best values are in bold. The second-best values are underlined. The baselines are statistically significant based on paired t-tests ($p < 0.05$) except for the values denoted with *(for ChatGPT-3.5) and †(for ChatGPT-4).

ChatGPT-4			ChatGPT-3.5		
Model	**Jaccard Index ↑**	**RBO ↑**	**Model**	**Jaccard Index ↑**	**RBO ↑**
Random†	0.0128	0.0136	Random	0.0091	0.0102
MostPop	0.0629	0.0975	MostPop	0.0464	0.0765
NeuMF	0.1128	0.1534	NeuMF	0.0717	0.1070
LightGCN	0.1183	0.1787	LightGCN	0.0787	0.1349
AttributeItemKNN	0.1521	0.2495	AttributeItemKNN	0.0990	0.1908
VSM	0.1582	0.2578	VSM	0.1015	0.1935
$EASE^R$	0.1594	0.2480	$EASE^R$	0.1078	0.2010
UserKNN	0.1994	0.3292	AttributeUserKNN	0.1248	0.2420
RP^3_β	0.2016	0.3417	UserKNN	0.1249	0.2469
AttributeUserKNN	0.2029	0.3290	RP^3_β	0.1279	0.2657
ItemKNN	0.2091	0.3426	ItemKNN	0.1313	0.2507
MF2020	0.2180	0.3479	MF2020	0.1349	0.2609
ChatGPT-3.5	**0.2563**	**0.4588**	ChatGPT-4	**0.2563**	**0.4588**

based on both Jaccard and RBO metrics. ChatGPT-3.5 also shows the highest similarity with ChatGPT-4 and, similarly, shares similarities with $EASE^R_{(CF)}$, $LightGCN_{(CF)}$, and MostPop. For

Table 4
Comparative analysis of **Lists Similarity** between ChatGPT-3.5 and ChatGPT-4 with the baselines on **MovieLens**. Arrows indicate whether higher (↑) or lower (↓) values are desirable for each metric. Best values are in bold. The second-best values are underlined. The baselines are statistically significant based on paired t-tests ($p < 0.05$) except for the values denoted with *(for ChatGPT-3.5) and †(for ChatGPT-4).

ChatGPT-4			ChatGPT-3.5		
Model	**Jaccard Index ↑**	**RBO ↑**	**Model**	**Jaccard Index ↑**	**RBO ↑**
Random†	0.0112	0.0110	Random*	0.0074	0.0076
VSM	0.0215	0.0274	VSM*	0.0162	0.0195
AttributeItemKNN	0.0341	0.0402	AttributeItemKNN*	0.0218	0.0245
LightGCN	0.0419	0.0493	LightGCN	0.0231	0.0277
NeuMF	0.0776	0.1117	NeuMF	0.0438	0.0680
RP^3_β	0.1229	0.2067	RP^3_β	0.0679	0.1299
ItemKNN	0.1374	0.2110	ItemKNN	0.0745	0.1282
UserKNN	0.1411	0.2326	UserKNN	0.0751	0.1511
MF2020	0.1456	0.2245	MF2020	0.0777	0.1527
AttributeUserKNN	0.1461	0.2324	AttributeUserKNN	0.0790	0.1539
MostPop	0.1464	0.2080	MostPop	0.0822	0.1545
$EASE^R$	**0.1721**	**0.2524**	$EASE^R$	0.0921	0.1784
ChatGPT-3.5	0.1394	0.2401	ChatGPT-4	**0.1394**	**0.2401**

the **Last.FM** dataset, ChatGPT-4's recommendations align closely with ChatGPT-3.5, followed by $MF2020_{(CF)}$, $ItemKNN_{(CF)}$, and $AttributeUserKNN_{(CBF)}$. While ChatGPT-3.5 shows a similar pattern using the Jaccard metric, the RBO metric is led by $RP^3_{\beta(CF)}$ and $MF2020_{(CF)}$, indicating a different item ranking approach. For the **MovieLens** dataset, ChatGPT-4 shows unexpected Jaccard similarity with $EASE^R_{(CF)}$ and MostPop, with ChatGPT-3.5 trailing behind. However, the RBO metric reveals $EASE^R_{(CF)}$ as the most similar to ChatGPT-4, followed by ChatGPT-3.5. ChatGPT-3.5's lists show the highest similarity with those of ChatGPT-4 across both metrics, followed by $EASE^R_{(CF)}$ and MostPop.

Notably, the high mutual similarities between the ChatGPT models meet expectations. However, their pronounced affinity with Collaborative Filtering recommenders supports our hypothesis: GPT models' proficiency extends beyond recognizing relevant content, encompassing the ability to leverage latent collaborative information within these models.

4. Conclusion

The integration of ChatGPT into Recommender Systems (RSs) has garnered significant attention, leading to the exploration of various methods for incorporating these models into RS pipelines. This study evaluates the performance of ChatGPT-3.5 and ChatGPT-4 as RSs and re-rankers using a Role-Prompting strategy across three domains: Books, Music, and Movies.

Our experiments demonstrate ChatGPT's ability to effectively re-rank recommendation lists. Utilizing the nDCG metric, we observe improvements in recommendation performance after re-ranking the lists with ChatGPT, indicating its effectiveness in enhancing the relevance of

recommended items.

We also analyze the similarity between the recommendation lists generated by ChatGPT models and those produced by content-based and collaborative filtering RSs. Our findings reveal that ChatGPT models exhibit a higher degree of similarity to collaborative filtering recommendation lists, suggesting that these models can leverage latent collaborative information.

The outcomes of this study suggest that valuable information for recommendation tasks exists within the latent space of these language models. Future research will focus on designing RSs that effectively harness this latent collaborative information to improve overall recommendation performance.

These results open new avenues for future research. We plan to further explore the latent collaborative space within these models to enhance recommendation accuracy. Additionally, we aim to understand the underlying reasons behind ChatGPT's strong re-ranking capabilities and investigate content-based recommendation aspects. Ethical considerations regarding user data collection for training these powerful models also warrant further investigation.

In conclusion, this study highlights the potential of large language models like ChatGPT for Recommender Systems. Their deep integration and evolution to produce personalized recommendations hold significant promise for the future of the field.

Acknowledgements. The authors acknowledge partial support of the following projects: OVS: Fashion Retail Reloaded, Lutech Digitale 4.0, Secure Safe Apulia, Patti Territoriali WP1, BIO-D, and MOST - Centro Nazionale per la Mobilità Sostenibile. We also gratefully acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

Additionally, this work has been carried out while Giovanni Servedio was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with Politecnico Di Bari.

References

[1] M. H. A. Abdullah, N. Aziz, S. J. Abdulkadir, H. S. A. Alhussian, N. Talpur, Systematic literature review of information extraction from textual data: Recent methods, applications, trends, and challenges, IEEE Access 11 (2023) 10535–10562.

[2] R. Burke, *Recommender Systems: An Introduction*, by dietmar jannach, markus zanker, alexander felfernig, and gerhard friedrichcambridge university press, 2011, 336 pages. ISBN: 978-0-521-49336-9, Int. J. Hum. Comput. Interact. 28 (2012) 72–73.

[3] B. Mitra, N. Craswell, An introduction to neural information retrieval, Found. Trends Inf. Retr. 13 (2018) 1–126.

[4] F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, US, 2022.

[5] V. Paparella, V. W. Anelli, L. Boratto, T. D. Noia, Reproducibility of multi-objective reinforcement learning recommendation: Interplay between effectiveness and beyond-accuracy perspectives, in: J. Zhang, L. Chen, S. Berkovsky, M. Zhang, T. D. Noia, J. Basilico, L. Pizzato, Y. Song (Eds.), Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023, ACM, 2023, pp. 467–478. URL: https://doi.org/10.1145/3604915.3609493. doi:10.1145/3604915.3609493.

[6] G. M. Biancofiore, Y. Deldjoo, T. D. Noia, E. D. Sciascio, F. Narducci, Interactive question answering systems: Literature review, CoRR abs/2209.01621 (2022).
[7] Y. Zhang, X. Chen, Q. Ai, L. Yang, W. B. Croft, Towards conversational search and recommendation: System ask, user respond, in: CIKM, 2018, pp. 177–186.
[8] J. Gao, C. Xiong, P. Bennett, Recent advances in conversational information retrieval, in: SIGIR, 2020, pp. 2421–2424.
[9] V. Paparella, V. W. Anelli, F. M. Nardini, R. Perego, T. D. Noia, Post-hoc selection of pareto-optimal solutions in search and recommendation, in: I. Frommholz, F. Hopfgartner, M. Lee, M. Oakes, M. Lalmas, M. Zhang, R. L. T. Santos (Eds.), Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023, ACM, 2023, pp. 2013–2023. URL: https://doi.org/10.1145/3583780.3615010. doi:10.1145/3583780.3615010.
[10] Y. Maarek, Alexa and her shopping journey, in: CIKM, ACM, New York, NY, USA, 2018, p. 1.
[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: NeurIPS, 2020.
[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.
[13] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, J. Zhang, Chat-rec: Towards interactive and explainable llms-augmented recommender system, CoRR abs/2303.14524 (2023).
[14] Z. Yue, S. Rabhi, G. de Souza Pereira Moreira, D. Wang, E. Oldridge, Llamarec: Two-stage recommendation using large language models for ranking, CoRR abs/2311.02089 (2023).
[15] Y. Wang, Z. Jiang, Z. Chen, F. Yang, Y. Zhou, E. Cho, X. Fan, X. Huang, Y. Lu, Y. Yang, Recmind: Large language model powered agent for recommendation, CoRR abs/2308.14296 (2023).
[16] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. J. McAuley, W. X. Zhao, Large language models are zero-shot rankers for recommender systems, in: ECIR (2), volume 14609 of *Lecture Notes in Computer Science*, Springer, US, 2024, pp. 364–381.
[17] 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, in: RecSys, 2023, pp. 1126–1132.
[18] J. Liu, C. Liu, R. Lv, K. Zhou, Y. Zhang, Is chatgpt a good recommender? A preliminary study, CoRR abs/2304.10149 (2023).
[19] J. Zhang, K. Bao, Y. Zhang, W. Wang, F. Feng, X. He, Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation, in: RecSys, 2023, pp. 993–999.
[20] D. Di Palma, V. W. Anelli, D. Malitesta, V. Paparella, C. Pomo, Y. Deldjoo, T. D. Noia, Examining fairness in graph-based collaborative filtering: A consumer and producer perspective, in: IIR, volume 3448 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 79–84.
[21] A. C. M. Mancino, A. Ferrara, S. Bufi, D. Malitesta, T. D. Noia, E. D. Sciascio, Kgtore: Tailored recommendations through knowledge-aware GNN models, in: RecSys, 2023, pp. 576–587.

[22] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2016) 19:1–19:19.
[23] I. Cantador, P. Brusilovsky, T. Kuflik, Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011), in: RecSys, ACM, New York, NY, USA, 2011, pp. 387–388.
[24] Z. Cui, J. Ma, C. Zhou, J. Zhou, H. Yang, M6-rec: Generative pretrained language models are open-ended recommender systems, arXiv preprint arXiv:2205.08084 (2022).
[25] J. Lin, R. Men, A. Yang, C. Zhou, Y. Zhang, P. Wang, J. Zhou, J. Tang, H. Yang, M6: multi-modality-to-multi-modality multitask mega-transformer for unified pretraining, in: KDD, ACM, New York, NY, USA, 2021, pp. 3251–3261.
[26] S. Geng, S. Liu, Z. Fu, Y. Ge, Y. Zhang, Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5), in: RecSys, ACM, New York, NY, USA, 2022, pp. 299–315.
[27] X. Wu, H. Zhou, W. Yao, X. Huang, N. Liu, Towards personalized cold-start recommendation with prompts, arXiv preprint arXiv:2306.17256 (2023).
[28] J. Zhang, R. Xie, Y. Hou, W. X. Zhao, L. Lin, J.-R. Wen, Recommendation as instruction following: A large language model empowered recommendation approach, arXiv preprint arXiv:2305.07001 (2023).
[29] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, J. McAuley, Large language models as zero-shot conversational recommenders, in: Proceedings of the 32nd ACM international conference on information and knowledge management, 2023, pp. 720–730.
[30] W.-C. Kang, J. Ni, N. Mehta, M. Sathiamoorthy, L. Hong, E. Chi, D. Z. Cheng, Do llms understand user preferences? evaluating llms on user rating prediction, arXiv preprint arXiv:2305.06474 (2023).
[31] 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, ????
[32] J. Li, W. Zhang, T. Wang, G. Xiong, A. Lu, G. Medioni, Gpt4rec: A generative framework for personalized recommendation and user interests interpretation, in: eCom@SIGIR, volume 3589 of *CEUR Workshop Proceedings*, 2023.
[33] L. Wang, E. Lim, Zero-shot next-item recommendation using large pretrained language models, CoRR abs/2304.03153 (2023).
[34] S. Sanner, K. Balog, F. Radlinski, B. Wedin, L. Dixon, Large language models are competitive near cold-start recommenders for language- and item-based preferences, in: RecSys, 2023, pp. 890–896.
[35] Z. Li, Y. Chen, X. Zhang, X. Liang, Bookgpt: A general framework for book recommendation empowered by large language model, Electronics 12 (2023) 4654.
[36] L. Xu, J. Zhang, B. Li, J. Wang, M. Cai, W. X. Zhao, J. Wen, Prompting large language models for recommender systems: A comprehensive framework and empirical analysis, CoRR abs/2401.04997 (2024).
[37] V. W. Anelli, A. Bellogín, A. Ferrara, D. Malitesta, F. A. Merra, C. Pomo, F. M. Donini, T. D. Noia, Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation, in: SIGIR, ACM, New York, NY, USA, 2021, pp. 2405–2414.

[38] H. Steck, Embarrassingly shallow autoencoders for sparse data, in: WWW, ACM, New York, NY, USA, 2019, pp. 3251–3257.
[39] B. Paudel, F. Christoffel, C. Newell, A. Bernstein, Updatable, accurate, diverse, and scalable recommendations for interactive applications, ACM Trans. Interact. Intell. Syst. 7 (2017) 1:1–1:34.
[40] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: EC, ACM, New York, NY, USA, 2000, pp. 158–167.
[41] J. S. Breese, D. Heckerman, C. M. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: UAI, 1998, pp. 43–52.
[42] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: SIGIR, 2020, pp. 639–648.
[43] S. Rendle, W. Krichene, L. Zhang, J. R. Anderson, Neural collaborative filtering vs. matrix factorization revisited, in: RecSys, 2020, pp. 240–248.
[44] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in: WWW, 2017, pp. 173–182.
[45] T. D. Noia, R. Mirizzi, V. C. Ostuni, D. Romito, M. Zanker, Linked open data to support content-based recommender systems, in: I-SEMANTICS, ACM, New York, NY, USA, 2012, pp. 1–8.
[46] Z. Gantner, S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Mymedialite: a free recommender system library, in: RecSys, ACM, New York, NY, USA, 2011, pp. 305–308.
[47] H. Steck, Evaluation of recommendations: rating-prediction and ranking, in: RecSys, 2013, pp. 213–220.
[48] V. Paparella, D. Di Palma, V. W. Anelli, T. D. Noia, Broadening the scope: Evaluating the potential of recommender systems beyond prioritizing accuracy, in: RecSys, 2023, pp. 1139–1145.
[49] W. Wang, H. Yin, Z. Huang, Q. Wang, X. Du, Q. V. H. Nguyen, Streaming ranking based recommender systems, in: SIGIR, 2018, pp. 525–534.
[50] J. Jin, X. Chen, F. Ye, M. Yang, Y. Feng, W. Zhang, Y. Yu, J. Wang, Lending interaction wings to recommender systems with conversational agents, in: NeurIPS, 2023.
[51] N. C. Chung, B. Miasojedow, M. Startek, A. Gambin, Jaccard/tanimoto similarity test and estimation methods for biological presence-absence data, BMC Bioinform. 20-S (2019) 644.
[52] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Trans. Inf. Syst. 28 (2010) 20:1–20:38.