Conversational Recommender Systems based on Extracting Implicit Preferences with Large Language Models

Woo-Seok Kim[1], Wooseung Kang[1], Hye-Jin Jeong[1], Suwon Lee[1], Chie Hoon Song[2] and Sang-Min Choi[1,*]

[1]Department of Computer Science and Engineering, Gyeongsang National University, 501, Jinju-daero, Jinju-si, Gyeongsangnam-do, Republic of Korea

[2]Department of Management of Technology, Gyeongsang National University, 501, Jinju-daero, Jinju-si, Gyeongsangnam-do, Republic of Korea

Abstract

Conversational recommender systems (CRS) have gained significant attention for their ability to provide personalized recommendations through conversational interfaces. CRS are increasingly being used in various fields such as e-commerce, entertainment, and customer services by understanding user preferences and providing personalized recommendations. Large Language Models (LLMs) have potential in recommendation systems due to their ability to understand and generate text, as well as their generalization and reasoning capabilities. In this paper, we propose a novel method that leverages LLMs to extract implicit information from conversations and explicitly incorporate it into recommendations. Our approach focuses on extracting implicit information such as user-preferred categories from conversations and explicitly adding it to the recommendation processes to enhance performance. We utilized Reddit-movie dataset, which provides rich conversational data, to extract users' implicit preferred movie genres from conversations and explicitly incorporate this information into the conversation to recommend movies. Experimental results show that both GPT-3.5-turbo and GPT-4 models perform exceptionally well at identifying user preferences and providing accurate recommendations. These findings demonstrate that utilizing implicit information extracted from conversations can effectively enhance recommendation quality, highlighting the potential of LLMs in conversational recommender systems.

Keywords

Conversational Recommender Systems, Large Language Models, Implicit User Preference, Classification

1. Introduction

Conversational Recommender Systems (CRS) offer personalized recommendations by engaging in direct conversations with users through conversational interfaces. These systems typically utilize users' past behavior data, explicit feedback, and information gathered during the conversation to make recommendations [1, 2]. However, users' needs are complex and ever-changing, presenting a significant challenge in effectively understanding and adapting to them [3]. In CRS, it is required to not only detect the challenges but also an understanding of ambiguous and implicit preferences in user dialogue. Therefore, CRS must continuously identify both the explicit requirements and implicit preferences of users during the conversation to provide the most useful recommendations.

With the advancement of Large Language Models (LLMs), the models are being actively utilized in various fields [4, 5]. LLMs not only excel in understanding and generating text but also hold potential as recommender systems through their generalization and reasoning abilities. For instance, LLMs can be employed to generate new items that users might prefer by analyzing user reviews or conversation logs [6].

Sixth Knowledge-aware and Conversational Recommender Systems (KaRS) Workshop @ RecSys 2024, September 14–18 2024, Bari, Italy.

*Corresponding author.

kws7327@gmail.com (W. Kim); kung0615@gnu.ac.kr (W. Kang); cucumber_love@gnu.ac.kr (H. Jeong); leesuwon@gnu.ac.kr (S. Lee); chsong01@gnu.ac.kr (C. H. Song); jerassi@gnu.ac.kr (S. Choi)

0009-0002-8759-3728 (W. Kim); 0009-0009-4568-5887 (W. Kang); 0009-0005-1103-5481 (H. Jeong); 0000-0003-2603-1385 (S. Lee); 0000-0003-2818-3411 (C. H. Song); 0000-0001-5950-3081 (S. Choi)

Recently, methods for leveraging LLMs in the CRS domain have also been proposed [7]. Integrating LLMs into CRS can provide a more natural and flexible conversational interface, effectively uncovering users' hidden needs. For example, LLMs can precisely interpret ambiguous needs expressed by users during a conversation and provide more sophisticated recommendations based on this understanding [3].

In this paper, we propose a novel method that converts users' implicit preferences within conversations into explicit ones using LLMs. Additionally, we use the converted explicit information as labels to create a multi-label classification model. We extract user preferences from conversations to explicitly identify categorical information. we then use a classification model to quantify this information, effectively reconstructing the conversations. The reconstructed conversations include explicit preference information. By leveraging both categorical and quantitative information, we enhance recommendation accuracy.

Based on our approach, we can make revelation for the hidden preferences contained in a user's conversations. Moreover, we leverage extracted information and LLM to show that more accurate recommendations can be derived when explicit information such as numerical preferences are used in the conversation. We also show the experiment results that the integration of CRS and LLM is a crucial step towards the development of user-centric recommendation systems and suggests possibility for future research.

The contributions of this paper are as follows:

- We propose a method to explicitly extract implicit preferences within conversations.
- We further suggest using the extracted preferences to design a multi-label model that quantifies categorical data.
- Experimental results demonstrate that our proposed approach significantly enhances the performance of CRS.

2. Related Work

In general, CRS understands the context and flow of conversations to offer appropriate recommendations without explicit user statements [2]. This understanding is achieved using natural language processing (NLP) techniques and reinforcement learning methods [8, 9]. By analyzing user responses in real-time, CRS enhances recommendation accuracy and continuously learns to offer increasingly personalized suggestions.

LLMs can be used to analyze user conversations in CRS since LLMs takes textual information as input and outputs related text. Because of this reason, recently, research on utilizing LLMs for recommendation systems has garnered significant attention [4, 6]. LLMs, trained on vast amounts of textual data, exhibit advanced language understanding and generation capabilities, performing exceptionally well across various domains. In recommendation systems, LLMs can deeply understand users' linguistic expressions and contexts, enabling more sophisticated recommendations. For example, LLMs analyze user-written reviews or posts to discern preferences and provide personalized recommendations accordingly [10]. Additionally, LLMs leverage various forms of textual metadata related to recommended items, effectively addressing the cold start problem for new users or items with sparse initial data [11, 12].

Studies on integrating LLMs into CRS cover various aspects. LLMs excel in understanding conversational contexts and discerning user intentions. By utilizing these capabilities, CRS can offer more natural and human-like conversational interfaces. For example, CRS powered by LLMs can detect subtle emotional shifts or changes in user interests during conversations and provide corresponding recommendations in real-time [3]. Recent research has proposed methods to further personalize user interactions through LLMs, incorporating real-time feedback to enhance recommendation accuracy and user satisfaction. These studies contribute to overcoming the limitations of traditional recommendation systems by integrating the robust language understanding capabilities of LLMs, ultimately delivering a superior user experience. In this paper, we leverage the characteristics of these LLMs to extract features containing user preferences from dialogues. Furthermore, we propose a recommendation model that reflects the extracted features.

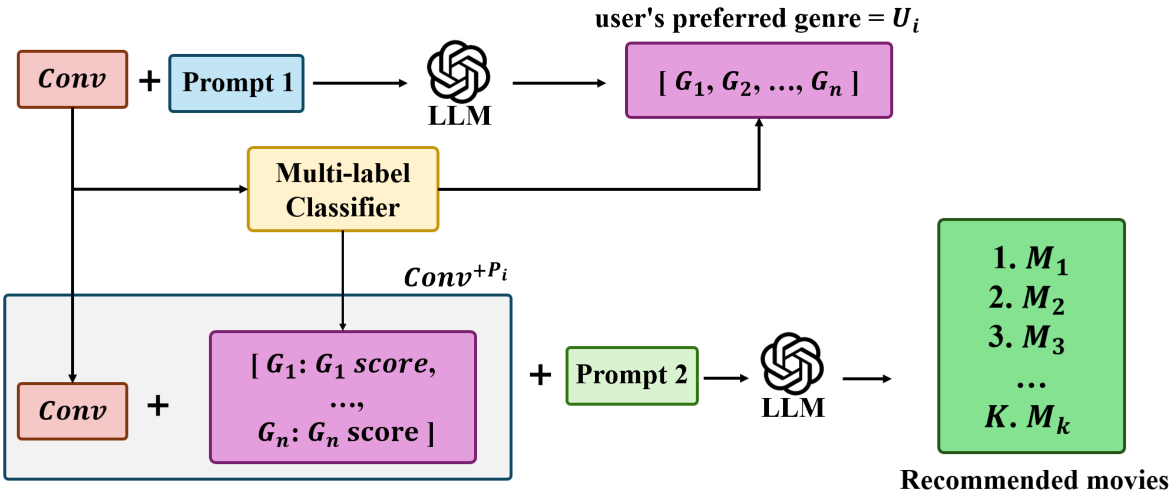
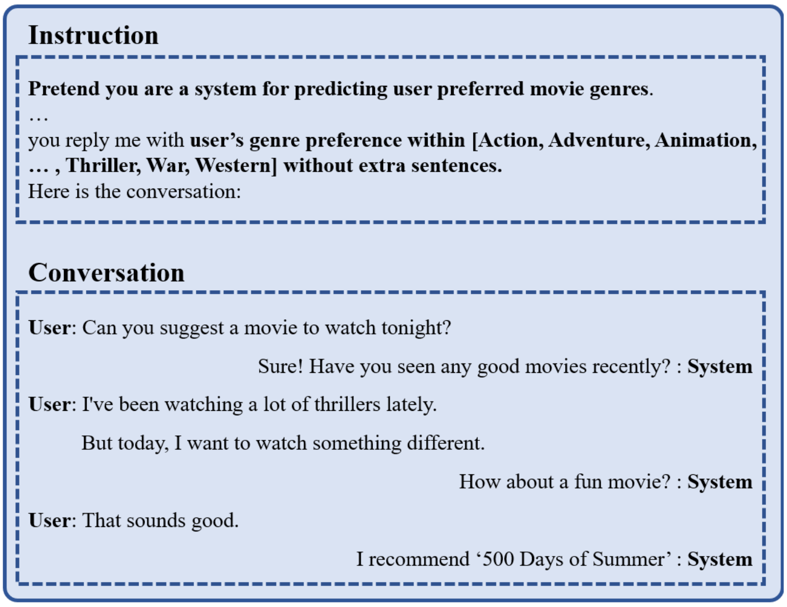


Figure 1: Overview of our approach

3. Our Approach

In this section, we propose a method to utilize users' implicit preferences explicitly. Fig. 1 shows overall processes of our approach. In this example, we utilize the movie domain. First, we extract the user's preferred genres implicitly expressed within the conversations using the LLM. Then, we train a multi-label classifier using the user's conversations as input and the extracted genres as labels. When the trained classifier receives the conversation as input, it outputs the predicted genre labels. These labels are then explicitly added to the end of the conversation, which is inputted into the LLM along with a prompt for movie recommendations. Finally, the LLM recommends a list of movies based on the prompt.

Our methodology consists of three main steps. The first step is to extract the user's preferences from the conversation and the second step is to train a multi-label classification model using the conversation and the extracted preferences. The last step is to make recommendations using the conversation and the classification model.



Instruction

Pretend you are a system for predicting user preferred movie genres.
…
you reply me with **user's genre preference within [Action, Adventure, Animation, … , Thriller, War, Western] without extra sentences.**
Here is the conversation:

Conversation

User: Can you suggest a movie to watch tonight?

Sure! Have you seen any good movies recently? : **System**

User: I've been watching a lot of thrillers lately.

But today, I want to watch something different.

How about a fun movie? : **System**

User: That sounds good.

I recommend '500 Days of Summer' : **System**

Figure 2: Prompt for extraction of implicit preferences within conversation

3.1. Extraction of Implicit Preferences within Conversation

We define user conversation as $Conv$ and the preference information extracted by the LLM from $Conv$ as U_i. Namely, U_i represents the item features, such as genres, that are positively expressed by a user i in $Conv$. For example, if a user i shows a positive reaction to a particular movie genre in the conversation, that genre can be considered as U_i.

To leverage LLM with extracted preferences, we configure the prompts to be suitable for extracting preferences. Prompts are generated by combining the user's conversation content with instruction. The prompt is structured as shown in Fig. 2. For instance, an instruction such as *"you reply me with user's genre preference within [Action, Adventure, Animation, … , Thriller, War, Western]"* is included in the prompt along with the conversation ($Conv$). Based on this prompt, the LLM extracts the genre (U_i) that the user is likely to prefer. The movie genres referenced are based on the genre list from IMDb [1], which includes 25 genres.

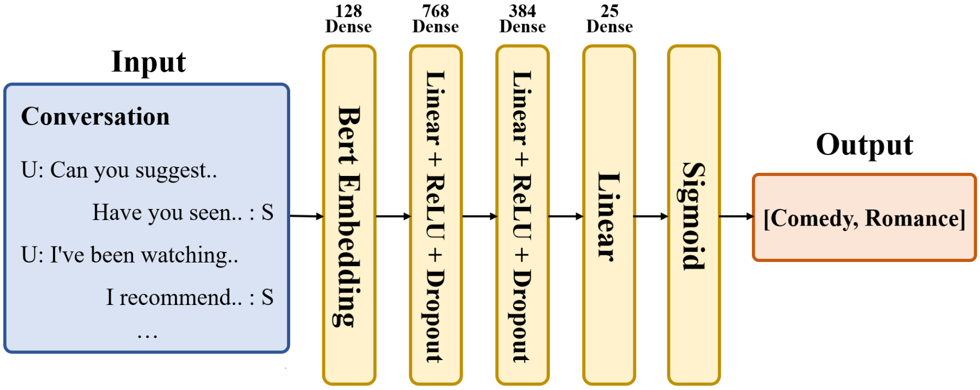


Figure 3: Architecture for multi-label classification model

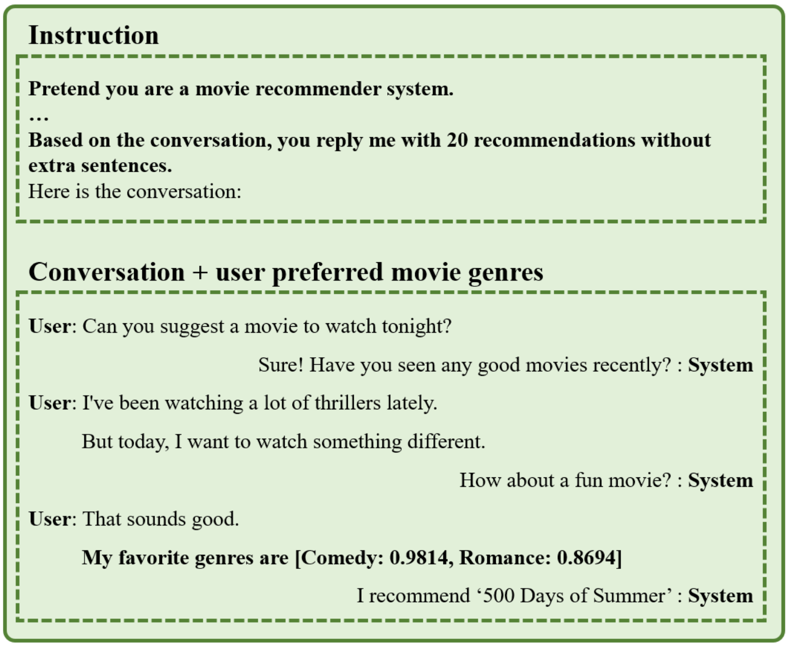
3.2. Quantifying Extracted Preferences

We propose a method to quantify extracted preferences applying a multi-label classification model. Using the conversation ($Conv$) as input and the user's preferred genre (U_i) as labels, a multi-label classification model is created. The model architecture is shown in Fig. 3. we first utilize BERT [13] to embed the conversation. BERT embedding converts the $Conv$ into vector form, making it understandable for the model. These embedded values pass through three linear layers, each further refining the understanding of the conversation and extracting important features. Finally, the output is generated through a sigmoid layer with 25 dense units representing the number of genres. This layer outputs probability values for each genre to predict the user's preferred genres.

Labels with predicted probabilities exceeding the threshold are defined as the quantification of user i's preferred genre P_i. We suppose that if we can quantify the extracted preference, it can help in clearly understanding the user's preferences through the comparison among the features. Because of this reason, P_i is designed to enable quantitative comparison by adding a numerical variable, the probability, to the categorical values of U_i. For example, if the user's preferred genres are predicted as [Romance, Comedy], this value can be included in the conversation to clearly indicate how much the user prefers these genres numerically.

To utilize the P_i in CRS, we explicitly add P_i to the $Conv$. For instance, it can be added as follows: *"My favorite genres are [Comedy: 0.9814, Romance: 0.8694]."* This restructured conversation is termed $Conv^{+P_i}$, indicating the original conversation ($Conv$) with the user's preferred genres (U_i) and their prediction probabilities explicitly added. It enables the user to clearly understand how much their preferences are reflected.

[1] https://www.imdb.com/



Instruction

Pretend you are a movie recommender system.
…
Based on the conversation, you reply me with 20 recommendations without extra sentences.
Here is the conversation:

Conversation + user preferred movie genres

User: Can you suggest a movie to watch tonight?

Sure! Have you seen any good movies recently? : **System**

User: I've been watching a lot of thrillers lately.
But today, I want to watch something different.

How about a fun movie? : **System**

User: That sounds good.
My favorite genres are [Comedy: 0.9814, Romance: 0.8694]

I recommend '500 Days of Summer' : **System**

Figure 4: Prompt for movie recommendations

3.3. Recommendation Process

Finally, we construct the prompt to be used as input for the LLM. The prompt consists of instruction and conversation containing user's preferred genres. The instruction composes sentences designed to instruct the LLM to recommend the top-k movies, followed by user conversation. The conversation also includes the predicted user's preferred genres (P_i). Using the prompt, the LLM can recommend the top-k movies based on the user conversation that include explicit preferences.

The prompt for movie recommendations is shown in Fig. 4. For example, the instruction specifies the role of LLM as a recommendation system tasked with recommending a total of 20 movies, along with details such as the number of recommendations. In addition, $Conv^{+P_i}$ includes information from the previous process, such as [Comedy: 0.9814, Romance: 0.8694], and the user's requirements. In response to the prompt, LLM recommends 20 movies that are slightly more focused on comedy than romance.

4. Experiments

We conduct the experiments using the GPT-3.5-turbo and GPT-4 models and the Reddit-movie datasets [7]. We address the datasets to reconstruct $Conv^{+U_i}$ and $Conv^{+P_i}$, and the performance of movie recommendations is compared using these two types of conversation data.

4.1. Experimental Setup

4.1.1. Datasets and Evaluation Metrics

We address the Reddit-movie datasets based on actual user conversations from Reddit, showcasing the personalized tendencies of users. It includes a variety of conversations and personalized preferences, making it suitable for CRS. We utilize the metrics Recall@K, NDCG@K, and MRR@K to evaluate the performance in our experiments [14]. We set the value of K as 1, 5, 10, and 20. These metrics are widely used indicators for evaluating the performance of recommendation systems, measuring the accuracy of the model at each K value.

- `Recall@K`: Measures the probability that the actual preferred genre is included in the top-k recommendations.

- NDCG@K: An indicator used to evaluate the quality of the recommendation list, considering the order of the recommended items.
- MRR@K: Based on the rank of the first correctly recommended item, with higher ranks receiving higher scores.
  Through these metrics, the performance of the movie recommendation lists generated by each GPT model can be quantitatively evaluated.

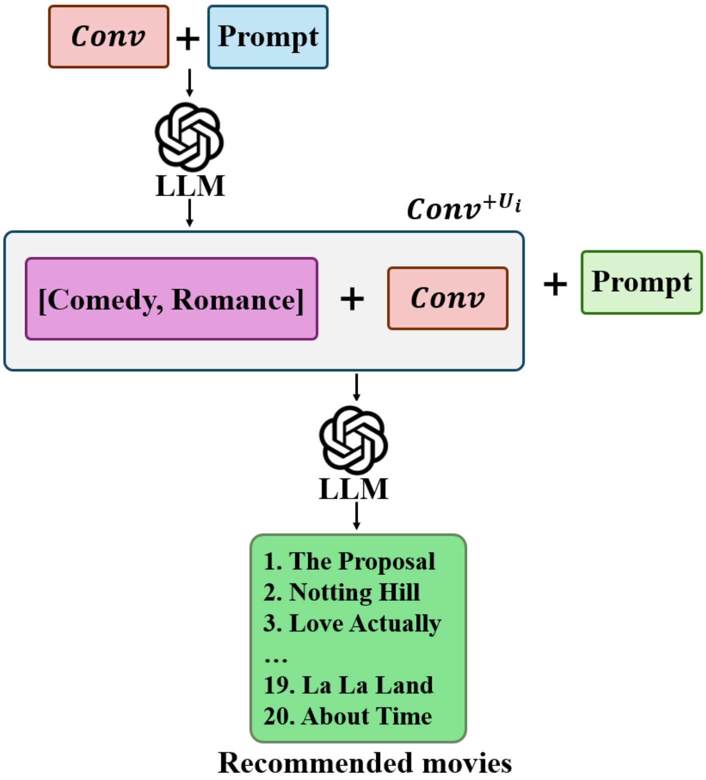


Figure 5: Example for the overview of experiments with $Conv^{+U_i}$

4.1.2. Baselines

The baseline for the experiment is to recommend movies that the user might prefer based on $Conv$ using LLM. The LLM uses a prompt consisting of $Conv$ and instructions to recommend 20 movies that the user might prefer. In the comparative experiment, $Conv$ is converted into $Conv^{+U_i}$ and $Conv^{+P_i}$, and the LLM recommends 20 movies in the same manner. For $Conv^{+U_i}$, as shown in Fig. 5, U_i is obtained through the LLM without using a multi-label classifier. Additionally, the prompt that converts $Conv$ to $Conv^{+P_i}$ includes the phrase *"The parentheses at the end of the conversation are in the format 'User's preferred genre: value'. The higher the value, the more preferred the genre."*

4.1.3. Implementation Details

The multi-label classifier is set up by using U_i generated by LLM as labels and $Conv$ as inputs to create models for each version of GPT. The training data is split into 80% for training and 20% for validation. The classifier is trained until the train loss and validation loss rates are below 0.10.

4.2. Experimental Results

The experimental results are summarized in Table 1 for each model. When we utilize the GPT-3.5-turbo model, we can observe that the performance of $Conv^{+U_i}$ surpasses that of using only $Conv$. However,

Table 1

Performance comparison of models. The best results are high lighted in bold

Model	Dataset	Recall@1	NDCG@1	MRR@1	Recall@5	NDCG@5	MRR@5	Recall@10	NDCG@10	MRR@10	Recall@20	NDCG@20	MRR@20
GPT-3.5-turbo	$Conv$	0.019955	0.019955	0.019955	0.070381	0.045299	0.037097	**0.106622**	**0.057018**	0.041931	**0.137435**	**0.064935**	0.044169
	$Conv^{U_i}$	**0.020786**	**0.020786**	**0.020786**	**0.072484**	**0.046781**	**0.038373**	0.103248	0.056663	**0.042413**	0.129659	0.06345	**0.044333**
	$Conv^{P_i}$	0.019906	0.019906	0.019906	0.069305	0.044663	0.03661	0.105497	0.056364	0.041437	0.135968	0.064197	0.043653
GPT-4	$Conv$	0.019906	0.019906	0.019906	0.069305	0.044646	0.03659	0.105497	0.056349	0.041417	0.136115	0.064219	0.043644
	$Conv^{U_i}$	0.019906	0.019906	0.019906	**0.06994**	0.045015	0.036868	0.106035	0.056685	0.041681	0.136604	0.064541	0.043904
	$Conv^{P_i}$	**0.019955**	**0.019955**	**0.019955**	**0.06994**	**0.045023**	**0.03688**	**0.106084**	**0.056711**	**0.041703**	**0.136897**	**0.064628**	**0.043941**

in other metrics, we can find that the performance of the original $Conv$ is superior.

In the case of $Conv^{+P_i}$, we can observe that all metrics show lower performance compared to $Conv$. For the GPT-4 model, we find that the performance of $Conv^{+P_i}$ exceeds that of all metrics. Similarly, $Conv^{+U_i}$ also shows improved performance in all aspects compared to $Conv$. Thus, we confirm that as the performance of the GPT model improves, explicitly expressing user preferences in conversations enhances the performance of the CRS. Furthermore, it is indicated that adding quantitative values to categorical data through the proposed multi-label classifier improves recommendation performance more than expressing preferences only categorically.

5. Conclusion

In this paper, we have proposed a method for improving the performance of conversational recommender systems (CRS) by extracting implicitly expressed user preferences from conversations using large language models (LLMs), and then explicitly adding these preferences. Additionally, we have suggested enhancing CRS performance by using a multi-label classifier to add quantitative values to categorical preferences. We have also highlighted the performance differences between the GPT-3.5-turbo and GPT-4 models, demonstrating that recommender systems leveraging the latest models yield better results. In particular, our experiments using the GPT-4 model showed that both $Conv^{+U_i}$ and $Conv^{+P_i}$ outperform the original $Conv$.

Despite the contributions of the recommendation performance, there still exists the possibility for improvement in several parts. Our study is confined to the Reddit-movie dataset. It is required to the additional research to validate the generalizability of the methodology across different domains and datasets, such as music, books, and food domains. Although we have used the GPT-3.5-turbo and GPT-4 models, further experiments with other LLMs or more advanced models are required. This can help identify performance differences across various models and determine the optimal one. Subsequent research can contribute to developing more sophisticated and versatile recommender systems applicable in a wider range of scenarios.

Acknowledgments

Following are results of a study on the "Leaders in Industry-university Cooperation 3.0" Project supported by the Ministry of Education and National Research Foundation of Korea (NRF), "Regional Innovation Strategy (RIS)" through the NRF funded by the Ministry of Education(MOE)(2021RIS-003), and the NRF grant funded by the Korea government (MIST) (No. RS-2022-00165785).

References

[1] K. Christakopoulou, F. Radlinski, K. Hofmann, Towards conversational recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 815–824. URL: https://doi.org/10.1145/2939672.2939746. doi:10.1145/2939672.2939746.

[2] Y. Sun, Y. Zhang, Conversational recommender system, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 235–244. URL: https://doi.org/10.1145/3209978.3210002. doi:10.1145/3209978.3210002.

[3] G. Zhang, User-centric conversational recommendation: Adapting the need of user with large language models, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1349–1354. URL: https://doi.org/10.1145/3604915.3608885. doi:10.1145/3604915.3608885.

[4] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, X. Xie, A survey on evaluation of large language models, ACM Trans. Intell. Syst. Technol. 15 (2024). URL: https://doi.org/10.1145/3641289. doi:10.1145/3641289.

[5] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, H. Xiong, E. Chen, A survey on large language models for recommendation, 2024. URL: https://arxiv.org/abs/2305.19860. arXiv:2305.19860.

[6] W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, C. Huang, Llmrec: Large language models with graph augmentation for recommendation, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 806–815. URL: https://doi.org/10.1145/3616855.3635853. doi:10.1145/3616855.3635853.

[7] 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, CIKM '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 720–730. URL: https://doi.org/10.1145/3583780.3614949. doi:10.1145/3583780.3614949.

[8] L. Zou, L. Xia, P. Du, Z. Zhang, T. Bai, W. Liu, J.-Y. Nie, D. Yin, Pseudo dyna-q: A reinforcement learning framework for interactive recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 816–824. URL: https://doi.org/10.1145/3336191.3371801. doi:10.1145/3336191.3371801.

[9] Y. Deng, Y. Li, F. Sun, B. Ding, W. Lam, Unified conversational recommendation policy learning via graph-based reinforcement learning, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1431–1441. URL: https://doi.org/10.1145/3404835.3462913. doi:10.1145/3404835.3462913.

[10] F. Yang, Z. Chen, Z. Jiang, E. Cho, X. Huang, Y. Lu, Palr: Personalization aware llms for recommendation, 2023. URL: https://arxiv.org/abs/2305.07622. arXiv:2305.07622.

[11] 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: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 890–896. URL: https://doi.org/10.1145/3604915.3608845. doi:10.1145/3604915.3608845.

[12] S. Agrawal, J. Trenkle, J. Kawale, Beyond labels: Leveraging deep learning and llms for content metadata, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1. URL: https://doi.org/10.1145/3604915.3608883. doi:10.1145/3604915.3608883.

[13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[14] A. Said, A. Bellogín, Comparative recommender system evaluation: benchmarking recommen-

dation frameworks, in: Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 129–136. URL: https://doi.org/10.1145/2645710.2645746. doi:10.1145/2645710.2645746.