Unveiling the Potential of LLMs in Simulated Society: A Knowledge-Driven LLM Agent Framework for User Modeling

Shengmao Zhu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zhushengmao23s@ict.ac.cn

Bingbing Xu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
xubingbing@ict.ac.cn

Yige Yuan
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
yuanyige20z@ict.ac.cn

Bin Xie
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
xiebin23s@ict.ac.cn

Yunfan Li
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
liyunfan24s@ict.ac.cn

Huawei Shen
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
shenhuawei.ict.ac.cn

Abstract

User modeling serves as the cornerstone of modern recommendation systems, focusing on the precise identification of user preferences and behavioral signatures to enable personalized service delivery. Existing recommendation methods face issues like sparse data and lack of transparency. The advent of large language models (LLMs) has brought new possibilities for inferring user preferences, which provides natural-language-based recommendation rationales and affective expressions. Motivated by AgentSociety Challenge @ WWW 2025 focused on simulating user-item interactions, we propose a knowledge-driven LLM agent framework for simulating user-item review interactions, which includes three modules: preference refinement, dual-signal injection, and category distinguisher. Specifically, preference refinement aims to model user and item profiles, enabling LLMs to perceive users and items comprehensively. Dual-signal Injection aims to incorporate external knowledge, such as collaborative filtering and distribution knowledge. Category distinguisher is designed to analyze the differences between true data and simulated data. Both online and offline experimental results demonstrate the effectiveness of this framework and prove the great potential of LLMs in user modeling.

CCS Concepts

• **Computing methodologies** → **Simulation environments**.

Keywords

LLM agent, User modeling, Knowledge-driven, Simulated society

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3719232

ACM Reference Format:
Shengmao Zhu, Bingbing Xu, Yige Yuan, Bin Xie, Yunfan Li, and Huawei Shen. 2025. Unveiling the Potential of LLMs in Simulated Society: A Knowledge-Driven LLM Agent Framework for User Modeling. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3719232

1 Introduction

User modeling serves as the cornerstone of modern recommendation systems, focusing on the precise identification of user preferences and behavioral signatures to enable personalized service delivery. The emergence of large language models (LLMs) has introduced novel capabilities in human-like preference inference, allowing computational agents to reconstruct latent user decision pathways through semantic reasoning. By orchestrating these agents within structured workflows, researchers now possess the means to systematically connect fragmented historical behaviors with specific prediction tasks, thereby advancing recommendation frameworks that balance predictive accuracy with operational transparency [9, 16].

This work is motivated by AgentSociety Challenge @ WWW 2025 focused on simulating user interactions within a controlled environment built from open-source datasets. Participants in the *User Modeling Track* are tasked with designing LLM-based agents to predict target users' star ratings and textual reviews for specific products. The agents operate in an interactive simulation, leveraging historical behavior, environmental data, and iterative feedback to refine their predictions.

Existing recommendation methods, such as collaborative filtering [2, 22] and matrix factorization [7, 8], often struggle with sparse data and cold-start scenarios. While neural architectures improve representation learning [12], they lack transparent decision pathways [15]. LLM-based agents fundamentally redefine this landscape through their unique capacity to articulate recommendation rationales in natural language formats. By generating textual explanations that explicitly highlight key phrases from user reviews, these agents provide sentiment-aware justifications that reveal their evaluation focus. More importantly, they can express nuanced emotional cues through lexical choices, effectively mirroring human evaluative patterns. This dual capability - combining explainable decision traces with affective expression - enables agents to construct personalized narratives that contextualize recommendations within users' emotional frameworks. Early implementations demonstrate superior performance in capturing subjective preferences through multi-modal analysis of review texts, social signals, and behavioral patterns, establishing agentic systems as transformative tools for human-centric recommendation design [6, 10, 20].

In this paper, we propose a knowledge-driven LLM agent framework for simulating user-item review interactions, which includes three modules: preference refinement, dual-signal injection, and category distinguisher. Specifically, preference refinement models user and item profiles for comprehensive perception, dual-signal injection integrates external knowledge like collaborative filtering, and category distinguisher analyzes true versus simulated data differences. Both online and offline experimental results demonstrate the effectiveness of this framework and prove the great potential of LLMs in user modeling, which achieves robust simulation of target users' review generation and rating prediction tasks while maintaining an average token consumption below 5,000 per inference instance.

2 Task

Our agent should simulate the preference and generate realistic review text based on the dataset given which contains three files including *item*, *user*, and *review*. There is both useful and redundant information contained in the dataset for the simulation task. The dataset and our task are briefly introduced as follows.

2.1 Dataset Overview

The dataset provided consists of *item*, *user*, and *review* with the sources are from three different platforms: Amazon, Yelp, and Goodreads. We observed that there is some redundant information contained in the dataset which are useless for the task. For the **item** file, this file contains information about the items being reviewed. The data from Amazon includes redundant information such as "images", "videos", and "bought_together". The data from Yelp includes redundant information such as "address", "city", "state", "latitude", "longitude", "is_open, postal_code", "hours", and "attributes". The data from Goodreads includes redundant information such as "link" and "URL". For the **user** file, this file contains information about the users. The data from Yelp includes redundant information such as "friends", "yelping_since", "elite". We need to clean this redundant information to ensure the dataset is efficient and effective for our simulation task. From **review** which contains the reviews provided by users, the information should be used to analyze users' historical evaluations and items' evaluation records, which will help in enhancing the Agent's simulation capabilities. The dataset used for evaluation consists of 40% simulated data and 60% true data.

2.2 Task Description

As a User ID and a certain Item ID are given for the task, the agent we constructed is required to model the user preference and generate realistic reviews based on the DATASET given to analyze historical user behavior records. For an input which includes User ID and Item ID, the output corresponded should include: A numerical score reflecting the simulated user's overall opinion as **Star**, and a detailed, contextually relevant commentary informed by user preferences and item attributes as **Review**. The metrics used to evaluate the performance of the agent are introduced as follow:

- **Preference Estimation** is calculated based on the star rating accuracy, using Mean Absolute Error defined as MAE = $\frac{1}{N}\sum_{i=1}^{N}|\hat{s}_{ni}-s_{ni}|$, where N is the total number of reviews in the evaluation set, $\hat{s}_{ni}$ is the normalized predicted star ration and s_{ni} is the ground truth star rating [14]. The final score of Preference Estimation is assigned as: $1 - \text{MAE}$.
- **Review Generation** is calculated based on three indicators including "Emotional Tone Error", "Sentiment Attitude Error" and "Topic Relevance Error". The Emotional Tone Error is a vector of emotion scores for the top five emotions in the review text is calculated using a predefined emotion classifier model [1], with each dimension normalized to the range $[0, 1]$. The Sentiment Attitude Error is analyzed using *nltk.sentiment.SentimentIntensityAnalyzer()*, with the resulting value normalized to the range $[0, 1]$. The Topic Relevance Error is an embedding vector for the review text is generated using a predefined embedding model [11], using cosine similarity between text embeddings to be the metric, measuring alignment with the real topics. We use MAE to evaluate the three indicators, and the final score of Review Generation is: $1-(\text{Emotional Tone Error}\times 0.25+\text{Sentiment Attitude Error}\times 0.25 + \text{Topic Relevance Error} \times 0.5)$.
- **Overall Quality.** Overall quality is calculated based on the preference estimation and review generation, assigned as: $(\text{PreferenceEstimation} + \text{Review Generation})/2$.

2.3 Challenges

The integration of LLM-based agents in personalized recommendation systems faces two fundamental tensions [3]: (1) **Balance Precision and Completeness in Information Extraction**: when processing vast amounts of historical review data, LLMs face a critical challenge: how to precisely extract decision-critical information from lengthy and complex reviews. Selecting only a subset of reviews may lead to the omission of vital information, while accepting everything indiscriminately could introduce noise that interferes with the model's judgment. (2) **Lack of Quantitative Standards**: although LLMs can identify the overall sentiment of users' reviews (e.g., positive or negative), they lack clear quantitative standards for specific scoring. For instance, a model might detect a positive sentiment but cannot accurately determine whether it corresponds to a rating of 3, 4, or 5. This deficiency in scoring granularity limits the potential of LLMs in fine-grained recommendations and personalized services.

3 Method

In this section, we first introduce the motivations underlying our architecture, followed by a comprehensive introduction of each modules in our knowledge-driven LLM agent framework.

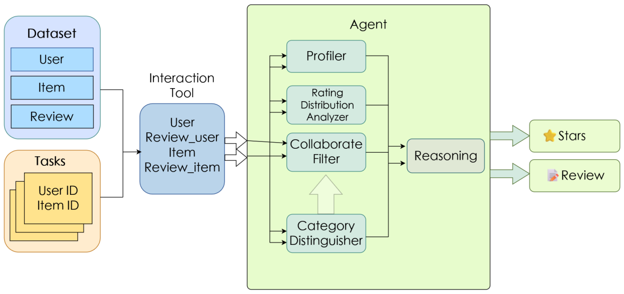


Figure 1: Overview of our agent framework, which includes three modules: preference refinement, dual-signal injection, and category distinguisher.

3.1 Motivation

Our architecture responds to the core tensions through two targeted innovations: (1) **Preference refinement.** By designing effective information extraction strategies, we can enable LLMs to identify critical insights from vast datasets of user reviews more efficiently [5]. This, in turn, enhances their accuracy and efficiency in decision-making across various applications, such as product recommendations and market analysis. Our preference alignment module is capable of accurately extracting key information from extensive historical data that aids the Agent in making informed decisions. This not only mitigates the interference caused by information overload but also highlight the crucial data, thereby laying a solid foundation for more precise user understanding. (2) **Knowledge Augmentation** [17]. Our knowledge augmentation provides a reference framework for the agent's rating prediction, enabling it to learn the quantitative patterns of user ratings. By mapping structured data and modeling probabilistic distributions, this mechanism helps establish an interpretable correspondence between semantic understanding and numerical judgment. As a result, the normativity and accuracy of rating decisions are significantly improved.

3.2 Modules

Inspired by the above motivation, we propose knowledge-driven LLM agent framework (see Figure **??**) to simulate user-item review interactions, which includes three modules as external knowledge to enhance LLM: preference refinement, dual-signal injection, and category distinguisher. Details of each modules are provided as follows.

3.2.1 *Preference Refinement.*

Item Profiler extracts discriminative features of the target item through systematic analysis of metadata from the *item* file and historical evaluation patterns from the *review* file. Through the analysis of the given dataset, we identify statistically significant attributes that exhibit strong correlations with user satisfaction levels , and discover latent user concerns through semantic pattern mining of review texts(e.g., "battery life" showing positive correlation with high ratings in electronics, "delivery delays" being recurring feature of negative reviews for perishable goods). These insights enable the Agent to dynamically adjust its simulation focus based on item category characteristics and historical evaluation patterns, significantly improving preference prediction accuracy. Prompts of Item Profiler are included in the Appendix A.

User Profiler constructs dynamic preference profiles through systematic analysis of the target user's review history from the *review* file in the dataset. By mining behavioral patterns in historical reviews of the target user, we identify statistically significant preference indicators that correlate with specific interaction behaviors(e.g., persistent emphasis on "ergonomic design" across positive reviews, recurring criticism of "overpriced accessories" in negative feedback). The profiling process automatically weights contextual biases and temporal dynamics during feature extraction. These refined user models enable the Agent to adapt its simulation strategy according to individual behavioral characteristics and temporal interaction patterns, substantially enhancing user preference simulation accuracy. Prompts used are included in the Appendix A.

3.2.2 *Dual-signal Injection.*

In this module, we inject collaborative filtering signals and rating distribution signals as dual signals into the LLM agent.

Collaborative Signal Injector [22]. This component enhances decision robustness through systematic integration of collaborative filtering signals from the *user* and *review* files. The user-based collaborative filtering (UBCF) [2] layer identifies behavioral neighbors by calculating cosine similarity across historical interaction patterns, with top-k similar users' ratings on target items formatted as contextual exemplars. While constrained by inherent data sparsity in real-world datasets, these collaborative signals provide complementary evidence that strengthens the LLM's prediction consistency when sufficient neighborhood information exists. As can be seen from Fig. 2, when our Agent uses collaborative filtering(CF)-based user reviews and CF scores as references for LLM, it can combine the strengths of both the original Agent and CF. In cases where either the Agent or CF performs suboptimally.

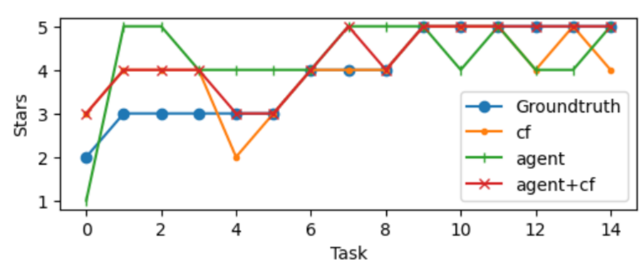
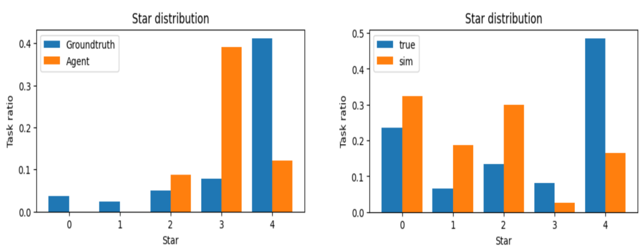


Figure 2: The star results under collaborative filtering, LLM Agent and their combination. The blue represents the star of ground-truth data.

Rating Distribution Analyzer [19]. We observed that the results from the agent exhibit a convergence in distribution, tending to assign moderate ratings or postive ratings, which is significantly different from the rating distribution of the ground truth as Fig. 3a. The phenomenon might result from three limitations in the agent: (1) the agent often over-considers positive or negative factors that are not important to the target user; (2) the LLM agent tend to give positive answers; and (3) the agent lacks a clear quantitative standard when assigning ratings, thus favoring neutral ratings. We address this issue by incorporating the historical rating distributions of the target user and target item as prior knowledge, and presenting it to the LLM in the form of a prompt. This would enable the LLM to analyze the rating criteria that the target user might apply when rating the target item. Specifically, this module generates multi-dimensional statistical profiles by analyzing quantitative patterns from the *review* file, including target user's historical rating distribution and item-specific rating trends. The derived metrics establish probabilistic boundaries for LLM reasoning while maintaining generative flexibility through adaptive confidence intervals. This mechanism enables the simultaneous preservation of statistical grounding and generative diversity in simulation outputs.

3.2.3 *Category distinguisher.*

We found that the rating distributions and the reviews of simulated tasks and real tasks have significant differences (see Fig. 3b). Also, the performance varies considerably between these two types of data. To distinguish them, we observed that the proportion of similar users to the target user who have interacted with the target item is significantly lower in simdata than in truedata. Utilizing this distinction, we apply different methods to these two data sources, which gain an improvement.



(a) Difference between groundtruth distribution and Agent score distribution.

(b) Stars distribution of tasks from simulated data and true data.

Figure 3: Star distribution analysis.

4 Optimization

In addition to the above modules, we have also applied several commonly used optimization strategies for LLMs, including in-context learning, review modified and memory mechanism.

4.1 In-Context Learning

ICL [4, 18] is a technique where a model is provided with a few examples of input-output pairs during inference, allowing it to adapt its predictions based on these examples. This approach leverages the provided examples to fine-tune its performance on the specific task at hand. By running offline evaluations, we identify *bad cases* the bad cases, i.e., predicted ratings deviate significantly from ground truth (>1-star difference). We then select representative challenging cases and generate failure explanations through semantic pattern analysis. The task descriptions, ground truth star ratings, reviews, and the generated explanations for the selected cases were compiled into examples, which were then included in the prompt provided to the Agent during inference. This allowed the Agent to learn from these specific cases and improve its performance on similar tasks.

4.2 Review Modifier

We implement review modifier to enhance the authenticity of generated reviews by aligning them with target users' historical writing patterns. Specifically, we extract user-specific linguistic signatures through lexical pattern mining and stylometric modeling from historical reviews of the target user. The modifier guides the LLM to rewrite the generated reviews, injecting extracted stylistic features while preserving content semantics. This mechanism enforces style consistency through contrastive decoding, maximizing alignment between generated reviews and authentic linguistic fingerprints.

4.3 Memory Mechanism

We implement memory modules [13, 21] to enhance the agent's capacity for dynamic knowledge reuse and preference adaptation. Specifically, we use a memory module to identify the most relevant historical interaction trajectory for a given query scenario and retrieves top-k candidate memories via vector similarity search, then employs LLM-powered semantic evaluation to assess their contextual alignment with the current task. We also design a memory module named Task Planning Memory to prompt the LLM to synthesize actionable strategies by analyzing successful trajectories when retrieving similar historical cases. And another module named Voyager Memory is used to optimize storage efficiency through trajectory summarization. We implement "item-review" key-value storage where similar items trigger relevant historical reviews from memory. Then we put the memory in the prompt as an example to guide the agent to generate. These modules leverage LLMs to enable context-aware memory storage, retrieval, and refinement, addressing challenges of information overload and decision traceability in user-item interaction simulations.

5 Experiment

We constructed the validation dataset comprising 40% simulated data (simdata) and 60% true data (truedata) for optimization. The simdata was from the data provided by the competition organizers. The truedata was selected by extracting the last evaluation behavior of a total of 300 users from three datasets. Table 1 presents the experimental results when different modules were utilized. The discrepancy between the online and offline results is likely due to the different proportions of various ratings within the two datasets.

Table 1: Experimental results with different modules. For the sake of brevity, M1 represents Item profiler. M2 represents User Profiler. M3 represents Collaborative Signal Injector. M4 represents Rating Distribution Analyzer. M5 represents Category distinguisher

Modules	Online Result	Offline Result
M1 & M2	0.8384	0.8339
M1 & M2 & M3 & M5	0.8423	0.8460
M1 & M2 & M3 & M4 & M5	0.8416	0.8533

6 Conclusion

In this paper, we introduce knowledge-driven LLM agent framework in details, which won the first place on the leaderboard.

Acknowledgments

This work was supported by the Strategic Priority Research Program of the CAS under Grant No. XDB0680302, and the National Natural Science Foundation of China (Grants No.U21B2046 and No.62202448).

References

[1] Francesco Barbieri, José Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. abs/2010.12421 (2020). https://api.semanticscholar.org/CorpusID:225062026

[2] Khadidja Belmessous, Faouzi Sebbak, M'hamed Mataoui, and Amine Batouche. 2022. Co-rating Aware Evidential User-Based Collaborative Filtering Recommender System. In *Communication Systems and Applications*. https://api.semanticscholar.org/CorpusID:257642071

[3] Junyi Chen. 2023. A Survey on Large Language Models for Personalized and Explainable Recommendations. *ArXiv* abs/2311.12338 (2023). https://api.semanticscholar.org/CorpusID:265308990

[4] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2022. A Survey on In-context Learning. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:255372865

[5] Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. 2024. Review-driven Personalized Preference Reasoning with Large Language Models for Recommendation. *ArXiv* abs/2408.06276 (2024). https://api.semanticscholar.org/CorpusID:271855183

[6] Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large Language Models meet Collaborative Filtering: An Efficient All-round LLM-based Recommender System. *ArXiv* abs/2404.11343 (2024). https://api.semanticscholar.org/CorpusID:269188358

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42 (08 2009), 30–37.

[8] T. Lekshmi Priya and Harikumar Sandhya. 2021. Matrix Factorization for Recommendation System. In *Advances in Artificial Intelligence and Data Engineering*, Niranjan N. Chiplunkar and Takanori Fukao (Eds.). Springer Nature Singapore, Singapore, 267–280.

[9] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *ArXiv* abs/2306.05817 (2023). https://api.semanticscholar.org/CorpusID:259129651

[10] Qiyao Peng, Hongtao Liu, Hua Huang, Qing Yang, and Minglai Shao. 2025. A Survey on LLM-powered Agents for Recommender Systems. https://api.semanticscholar.org/CorpusID:276395083

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:201646309

[12] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation Learning with Large Language Models for Recommendation. *Proceedings of the ACM on Web Conference 2024* (2023). https://api.semanticscholar.org/CorpusID:264439548

[13] Chengbing Wang, Yang Zhang, Fengbin Zhu, Jizhi Zhang, Tianhao Shi, and Fuli Feng. 2024. Leveraging Memory Retrieval to Enhance LLM-based Generative Recommendation. *ArXiv* abs/2412.17593 (2024). https://api.semanticscholar.org/CorpusID:274981574

[14] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. RecMind: Large Language Model Powered Agent For Recommendation. In *NAACL-HLT*. https://api.semanticscholar.org/CorpusID:261243746

[15] Colin White, Willie Neiswanger, Sam Nolen, and Yash Savani. 2020. A Study on Encodings for Neural Architecture Search. *ArXiv* abs/2007.04965 (2020). https://api.semanticscholar.org/CorpusID:220424581

[16] Likang Wu, Zhilan Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A Survey on Large Language Models for Recommendation. *ArXiv* abs/2305.19860 (2023). https://api.semanticscholar.org/CorpusID:258987581

[17] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. *ArXiv* abs/2306.10933 (2023). https://api.semanticscholar.org/CorpusID:259202547

[18] Hongling Xu, Qianlong Wang, Yice Zhang, Min Yang, Xi Zeng, Bing Qin, and Ruifeng Xu. 2024. Improving In-Context Learning with Prediction Feedback for Sentiment Analysis. *ArXiv* abs/2406.02911 (2024). https://api.semanticscholar.org/CorpusID:270257772

[19] An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024. On Generative Agents in Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 1807–1817. doi:10.1145/3626772.3657844

[20] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji rong Wen. 2023. Recommendation as Instruction Following: A Large Language Model Empowered Recommendation Approach. *ArXiv* abs/2305.07001 (2023). https://api.semanticscholar.org/CorpusID:258615776

[21] Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A Survey on the Memory Mechanism of Large Language Model based Agents. *ArXiv* abs/2404.13501 (2024). https://api.semanticscholar.org/CorpusID:269293320

[22] Yaochen Zhu, Liang Wu, Qilnli Guo, Liangjie Hong, and Jundong Li. 2023. Collaborative Large Language Model for Recommender Systems. *Proceedings of the ACM on Web Conference 2024* (2023). https://api.semanticscholar.org/CorpusID:264935253

A Prompts

```
There is a business that you might be interested to visit
     . Here is the information about the business: {item
     }. Others have reviewed this business before: {
     reviews_history}. Please analyze the advantages and
     disadvantages of this business based on the business
      information and user reviews.
Consider the following aspects:
1. What are the key features and benefits of the business
     ? What makes it stand out?
2. What are the potential drawbacks and limitations of
     the business? What are the common complaints?
3. Which aspects are most frequently mentioned in the
     reviews? The more frequently an aspect is mentioned,
      the more significant it is, whether it's a benefit
     or a drawback.
Requirements:
- Your analysis should be based on the provided business
     information and user reviews
- You don't need to provide any evidence or examples,
     avoid phrases like "as evidenced by" or similar
     expressions
- Pay more attention to the points that are mentioned
     multiple times
- Your response should be 2-3 sentences, focusing on the
     key benefits, drawbacks and providing a simple
     summary of the overall quality
```

Listing 1: Prompt for ItemProfiler

```
You are a real human user on yelp. Here is your review
     history for businesses you have interacted with: {
     reviews_history}. Please analyze your review history
      to determine your interests and preferences for
     businesses.
Consider the following aspects:
1. What are your interests and preferences? Or what are
     the advantages of the business that you pay more
     attention to?
2. What are your dislikes and pet peeves? Or what are the
      shortcomings of the business that will make you
     unbearable?
3. What are your key needs or what are you looking for
     when visiting a business? For example, price,
     quality, service, etc.
Requirements:
- Your analysis should be based on the provided review
     history
- Your response should be 1-3 sentences, focusing on the
     key aspects of your preferences
- Response in first-person using "I" statements
```

Listing 2: Prompt for UserProfiler