SMART Restaurant Recommender: A ContextAware Restaurant Recommendation Engine

Ayesha Ubaid *, Adrian Lie , Xiaojie Lin

Posted Date: 24 February 2025

doi: 10.20944/preprints202502.1821.v1

Keywords: Large language Models; LLMs; Open AI; ChatGPT; Restaurant Recommender System; Google API; Recommender Systems

 doi:10.20944/preprints202502.1821.v1

Article

SMART Restaurant Recommender: A ContextAware Restaurant Recommendation Engine

Ayesha Ubaid [1], Adrian Lie [2] and Xiaojie Lin [2]

[1] Artificial Intelligence Institute of Australia, Department of Computer Science, University of Technology Sydney, 15 Broadway Ultimo, Sydney, 2000, NSW, Australia

[2] Department of Computer Science, University of Technology Sydney, 15 Broadway Ultimo, Sydney, 2000, NSW, Australia

* Correspondence: ayesha.ubaid@uts.edu.au

Abstract With the rise of ecommerce systems and web application usage, recommendation systems have become important to our daily tasks. They provide personalized suggestions to assist with any task under consideration. While various machine learning algorithms have been developed for recommendation tasks, existing systems still face limitations. This research focuses on advancing contextaware recommendation systems by leveraging the capabilities of Large Language Models (LLMs) in conjunction with realtime data. The research exploits the integration of existing realtime data APIs with LLMs to enhance the capabilities of the recommendation systems already integrated into smart societies. The experimental results demonstrate that the hybrid approach significantly improves the user experience and recommendation quality, ensuring more relevant and dynamic suggestions.

Keywords: large language models; LLMs; Open AI; ChatGPT; restaurant recommender system; Google API; recommender systems

Introduction

The rapid advancement in artificial intelligence (AI) and natural language processing (NLP) have significantly transformed the way people access and engage with information. The emergence of large language models (LLMs), which represent a breakthrough in NLP, has significantly expanded the scope and effectiveness of recommendation systems within smart societies [1]. LLMs empower recommendation systems by enabling them to generate contextaware recommendations by accurately capturing user preferences, generating more personalized and diverse recommendations along with insightful and explainable recommendations [2]. However, LLMsbased recommendation systems suffer from the problem of discriminative recommendations. This means that the computational resources to calculate the ranking score are quite expensive due to their large platform [3].

Industrial recommendation systems are typically developed in multiple stages to narrow down the accurate candidate. Various approaches are being explored to simplify LLMbased recommendation architectures while maintaining high performance without incurring excessive computational costs. Notably, the integration of AI models with datarich sources presents novel opportunities for enhancing recommendation accuracy. This expanding digitization has shifted the interaction of humans with these systems. The same behavior expands the social interaction of humans and affects their social activities. Everyone is relying on the recommendations provided by the in formation systems. The same behavior can be seen when people try to find the restaurant of their choice. However, carefully considering both the potential benefits and associated risks is crucial to ensure the ethical use of this technology [4].

 doi:10.20944/preprints202502.1821.v1

Traditional restaurant recommendation platforms, such as Yelp and Trip Advisor, primarily rely on usergenerated reviews and ratings. While useful, these approaches often fail to capture nuanced contextual preferences such as dietary restrictions, desired ambiance, or cuisine preferences. Additionally, most existing recommendation systems struggle with interpreting complex user queries and providing realtime, personalized suggestions. In contrast, LLMs, such as ChatGPT, possess advanced natural language understand ing capabilities but lack live access to realtime restaurant data, including location, operating hours, reviews, and business status.

The integration of LLMs with realtime data frameworks, such as Google Places API, presents a promising avenue for enhancing restaurant recommend dation systems. This synergy enables more relevant and contextually aware suggestions by combining the interpretative power of generative AI with up todate, locationspecific information. As a result, users benefit from recommendations that are not only personalized but also dynamically adapted to real world changes.

This study investigates the cooperative potential of generative AI—specifically, LLMs—within existing recommendation frameworks and evaluates their efficacy in improving social experiences by providing personalized restaurant recommendations based on user context. However, a key challenge in de veloping such a system lies in designing effective prompts that enable LLMs to accurately interpret and respond to natural language queries, which are inherently ambiguous and contextdependent [5].

In the next Section 2 we have summarized the related work followed by the proposed research design, and development in Section 3. Section 4, explains the experimental and evaluation. Section 5 discusses the evaluation results followed by a conclusion and future work in Section 6.

Related Work

The exploration of LLMs and their interaction with Application Programming Interfaces (APIs) such as the Google Places API for providing recommendations is an emerging area of interest in natural language processing (NLP) and artificial intelligence (AI). This literature review synthesizes recent studies to outline current methodologies and challenges in how LLMs communicate with the Google Places API to provide dynamic and personalized recommendations.

The study conducted by Silva and Tesfagiorgis in 2023 [6] examined various prompt designs generated by GPT4 to enhance the effectiveness of LLMs when interacting with APIs. The authors conducted experiments to identify the most efficient prompt structures, finding that finetuned prompted LLMs performed significantly better than no fine tuned systems in terms of accuracy and response times. The research outlined the importance of optimizing prompt designs being inputted into LLMs. The research by Patil et al. [7] highlights the benefits of finetuning LLMs, demonstrating that models tailored to specific tasks consistently outperform general purpose models like ChatGPT4 and Llama3 in API interactions. The study tested and revealed results of improved performance metrics, such as precision and speed, when the LLMs were finetuned. Similarly, Luo et al. [8] investigated various finetuning techniques and their impact on LLM performance. By testing various methods, their study identified the most effective strategies for optimizing response times and ensuring relevance in API generated results. The findings confirmed that specific finetuning approaches enhance efficiency and accuracy, making LLMs more viable for real time applications requiring rapid data retrieval. Roumeliotis et al. [9] investigated the integration of LLMs with unsupervised learning techniques such as K-Means clustering and content based filtering to refine product recommendation systems. Their study demonstrated that incorporating GPT4's advanced natural language understanding significantly improved the precision and relevance of recommendations. Spinelli [10] examined the importance of robust language models in processing intricate queries efficiently, emphasizing that advanced linguistic comprehension is essential for accurate API interactions. This research supports the use of ChatGPT4, an LLM already capable of processing com plex language structures, to communicate effectively with the Google Places API. The research by Mao et al. [11] proposes that larger LLMs can handle diverse data inputs efficiently but may be subdued to potential stability risks due to adaptive learning techniques. The paper

 doi:10.20944/preprints202502.1821.v1

discusses how scalable approaches might lead to instability in the models, suggesting that while scalability is essential, it must be balanced with measures to maintain system stability and reliability. Lin et al. [12] provided an in-depth analysis of LLMs in recommender systems, outlining their roles in different stages of the recommendation pipeline, such as feature engineering and user interaction. Their study also addressed key challenges, including efficiency, effectiveness, and ethical concerns when using LLMs in recommendation tasks. Hu et al. [13] indicates that scalable training strategies may compromise the overall efficiency of LLMs, similar to what was said in the research paper by Mao et al. [11]. The study discusses the balance between adaptability and stability, suggesting careful consideration of training strategies to ensure that scalability does not come at the cost of efficiency and performance. Nathan et al. [14] in his work presented a framework that was able to integrate LLMs with traditional reinforcement learners. He experimented with this integration both in the context of movies and book recommendation settings. In [15], the authors have reviewed latest LLMs and their integration with existing recommendation systems along with the fine-tuning techniques. The author has discussed whether finetuning of whole LLM can be done. However, fine tuning of the prompt can perform the same task to avoid the pain. The main focus of the research was to work on prompt design effectively to leverage the effectiveness of the large language models. The reviewed literature revealed gaps in understanding the long-term learning capabilities of LLMs and their adaptation to dynamic, real time data. Most studies focused on specific API tasks such as email automation or weather forecasting, where static or minimally changing data was used. However, limited research has explored how LLMs handle real time, dynamic data interactions—particularly in the con text of recommendation systems. Furthermore, there were no peer reviewed research papers on LLMs specifically interacting with the Google Places API to provide recommendations based on user input.

Table 1. Comparison Among Existing Works.

Author(s)	Key Findings.
Silva & Tesfagiorgis (2023) [6]	Effective prompt designs & finetuning methods.
Patil et al. (2023) [7]	Finetuned LLMs outperform GPT4 in API inter actions.
Luo et al. (2024) [8]	Finetuning improves LLM performance w.r.t. response times.
Roumeliotis et al. (2024) [9]	LLMbased unsupervised clustering enhances recommendation precision.
Spinellis (2024) [10]	Linguistic structures are crucial for precise API
calls. Mao et al. (2024) [11]	LLMs handle diverse inputs efficiently but risk stability.
Lin et al. (2023) [12]	Examined LLMs in recommendation pipelines, focusing on efficiency and ethical considerations.
Hu et al. (2024) [13]	Scalable training strategies might compromise LLM efficiency.
Nathan et al. & Giorgio et al. (2024) [14]	Integrated LLMs to improve RLbased recommendations.
Fan et al. (2023) [15]	Emphasized the finetuning of prompt design to leverage the effectiveness of the LLMs for context aware recommendations.

Currently, ChatGPT's recommendations are constrained by its inability to access live data while recommending contextually aware restaurant rec ommendations such as operating hours, reviews, and business status. This research project aims to address this limitation by integrating the Google Places API with ChatGPT4o to provide contextaware recommendations based on user queries and realtime data.

Proposed Framework: GPT Restaurant Recommender

In this research, we have developed a framework that enables the seam less integration of LLMs with existing recommendation systems to provide context sensitive and personalized recommendations. The development of such a framework paves the way for future advancements in

 doi:10.20944/preprints202502.1821.v1

recommendation systems by leveraging real time data to address existing research gaps. To achieve this, we have designed and developed the restaurant recommendations system utilizing ChatGPT4.o and Google API. The proposed system offers recommendations based on user defined criteria, including dietary preferences, desired ambiance, budget constraints, and location.

1.1. Framework Design

The designed framework consists of three layers as shown in Figure 1. The first layer is UI layer. This layer allows user to interact with the syst em and submit query. The same layer is responsible for sending context tualized recommendations back to the user. This layer has been developed using ShadcnUI components [16], chosen for their flexibility and ability to deliver a clean, responsive user experience. The Google Maps Embed feature via ext.js [17] provides users with an interactive map view of recommended restaurants. The intermediate layer is available to processing query and pro vide recommendations while accessing the live data which is then passed to the third layer for further optimization using LLMs. Once the LLM processes the query's context, it sends the recommendations back to the UI layer. The backend is implemented using Next.js, which provides API routes to manage requests and data flow between the user interface, Google Places API, and OpenAI API [18]. The aisdk/Open AI library enables direct interaction with OpenAI's ChatGPT4o model. The AI Client class in the backend manages the recommendation generation, using the custom prompt schema to provide ChatGPT with structured data.

To materialize the effectiveness of the proposed framework, a web application named "GPT restaurant recommender" is developed. The technology stack adopted for this development includes Next.js [19] deployed on Ver cel [20], a modern deployment platform optimized for serverless web applications. The architecture integrates several components to deliver recommendations. Figure 2 shows the system architecture diagram.

The user initiates the process by entering their inquiry such as location, dietary requirements, and any other additional context to the web application. The system makes an API call to Google Places to collect initial restaurant data based on user preferences. The Google Places API returns a list of restaurants that match the initial query parameters, providing information such as restaurant name, location, rating, and other relevant attributes. ChatGPT processes the restaurant data retrieved from Google Places alongside the user's specific criteria. ChatGPT evaluates each restaurant to determine the top recommendations based on the relevance of the user's query. Once the evaluation is complete, ChatGPT selects the top three relevant restaurants from the list, by filtering based on user preferences and embedded ethical constraints.

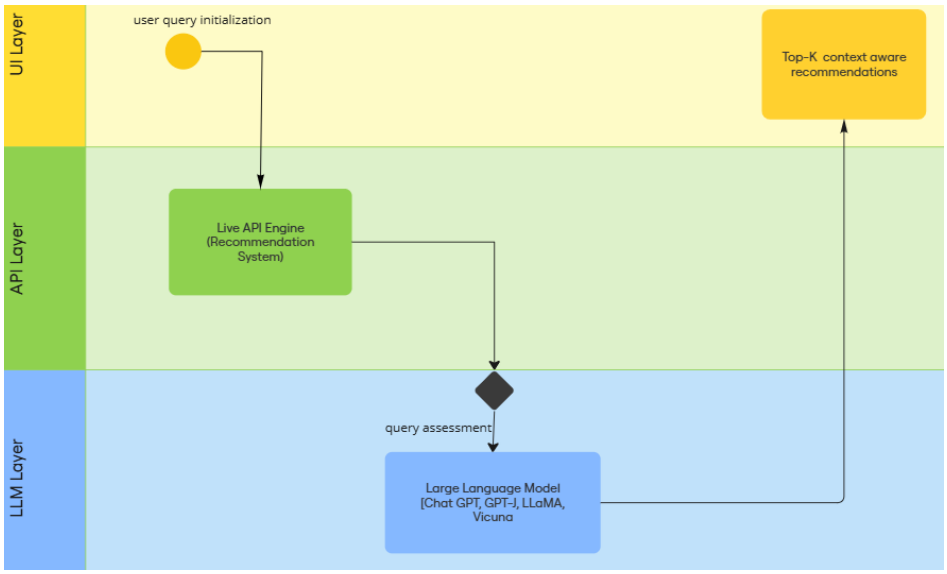


Figure 1. Framework for LLMs Enabled Recommendation Systems.

 doi:10.20944/preprints202502.1821.v1

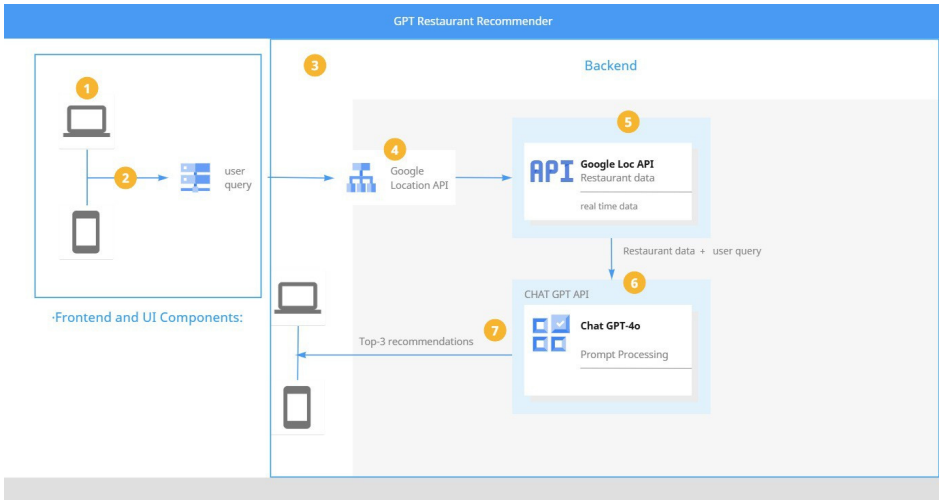


Figure 2. Architecture Diagram of GPT Restaurant Recommender.

The selected restaurants are then sent to users via the web interface, each accompanied by details such as location, contact information, and ratings. The user views the final recommendations, allowing them to choose a restaurant based on the personalized suggestions provided. This structured flow ensures that the system meet user requirements for personalized, location based, and relevant restaurant recommendations while maintaining quick response times. Figure 3 illustrates the system process for the diagram. The screen grabs of the web application to realize the research idea are shown in Figures 4, 5 and 6 respectively. The web application begins by prompting the user to enter their preferences to receive personalized restaurant recommendations, as shown in Figure 4a. Upon clicking "Find restaurants" button, the system initiates a searching process (Figure 4b), retrieving and ranking relevant restaurant options. The top three recommendations are then displayed to the user. When a restaurant is selected, additional details such as rating, budget indication and an interactive map are proved to the user for convenience as shown in Figure 4c. Figure 5a demonstrates a search performed for the Sydney region. Clicking on the map feature allows users to view the restaurant's precise location on Google Maps, offering seamless navigation assistance as shown in Figure 5b.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 February 2025 doi:10.20944/preprints202502.1821.v1

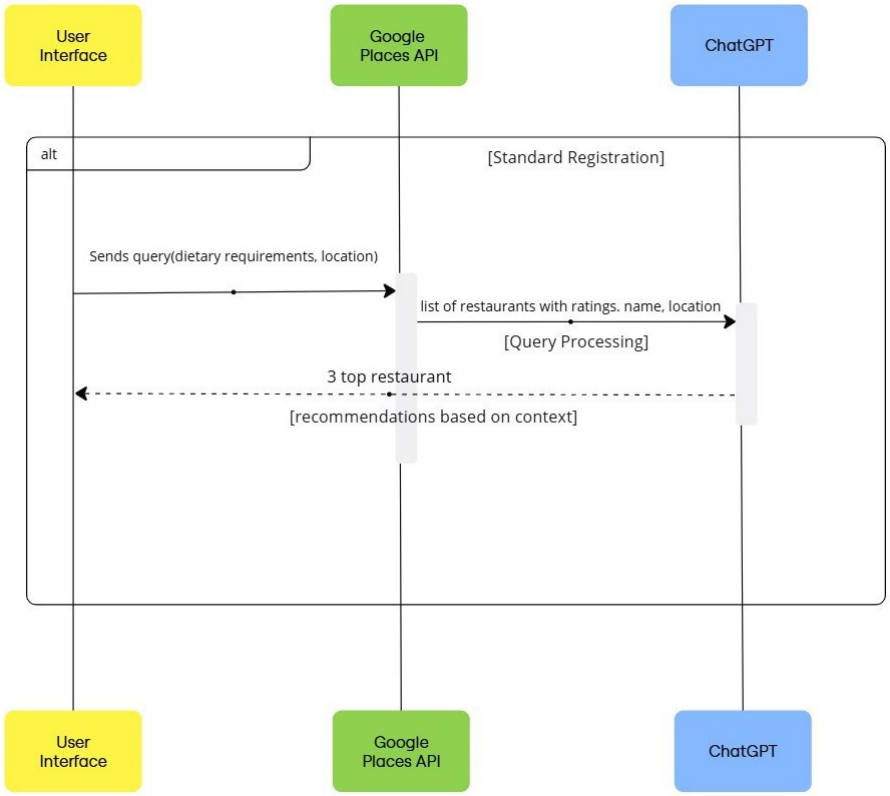
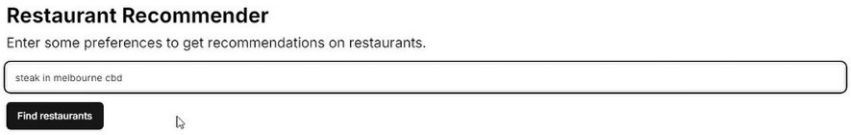
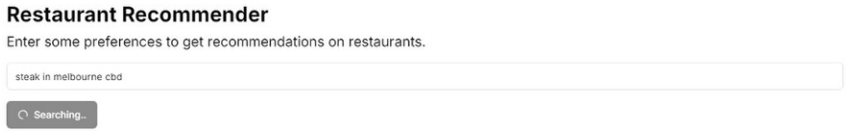


Figure 3. System Sequence Diagram of GPT Restaurant Recommender.

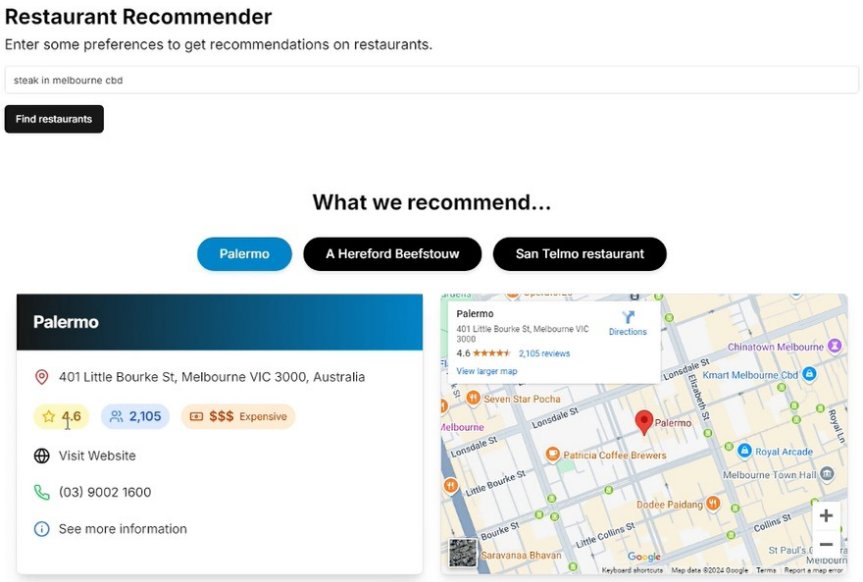


(a) Search Bar for Preferences of Restaurant Recommender



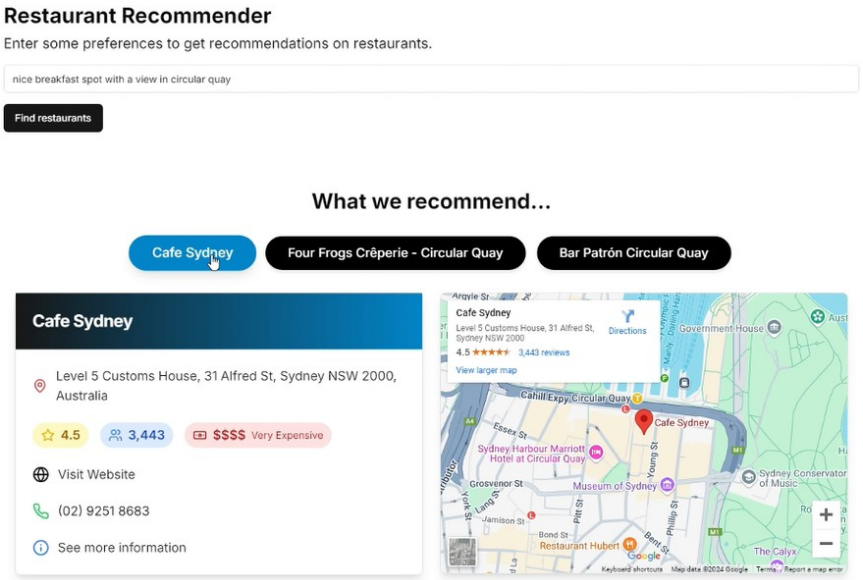
(b) Recommendation Results of Restaurant Recommender

 doi:10.20944/preprints202502.1821.v1

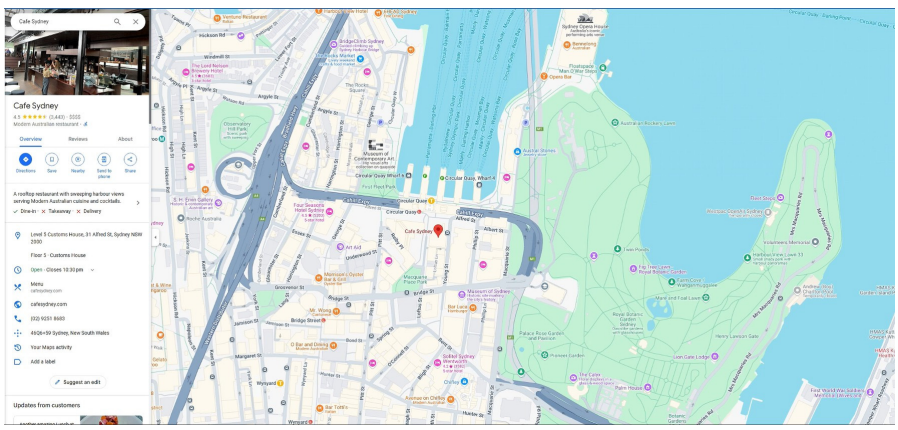


(c) Details of Selected Restaurant of Restaurant Recommender

Figure 4. Web Application of Smart Restaurant Recommender.



(a) Map Linked to Suggested Restaurant

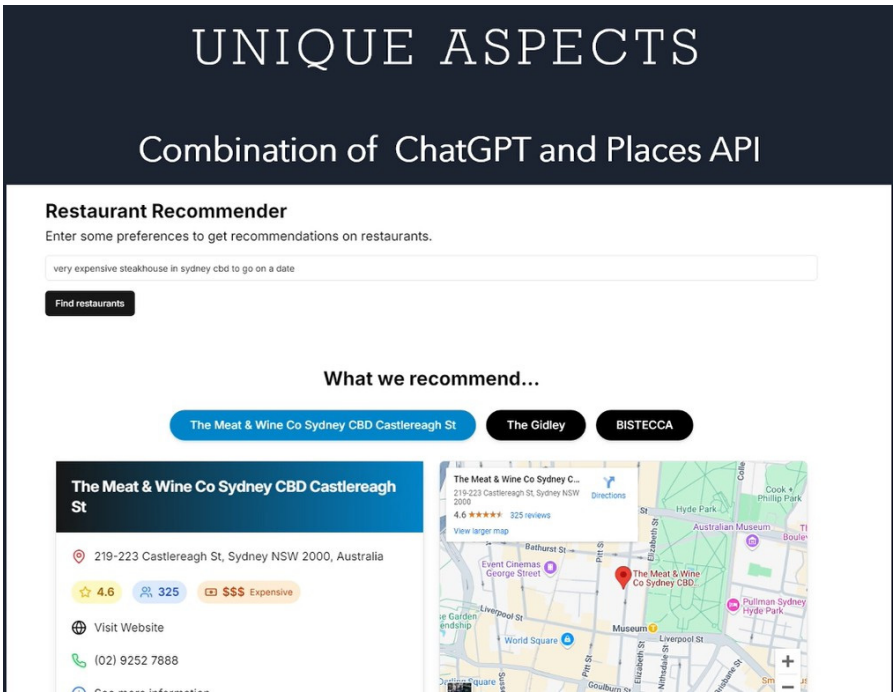


 doi:10.20944/preprints202502.1821.v1

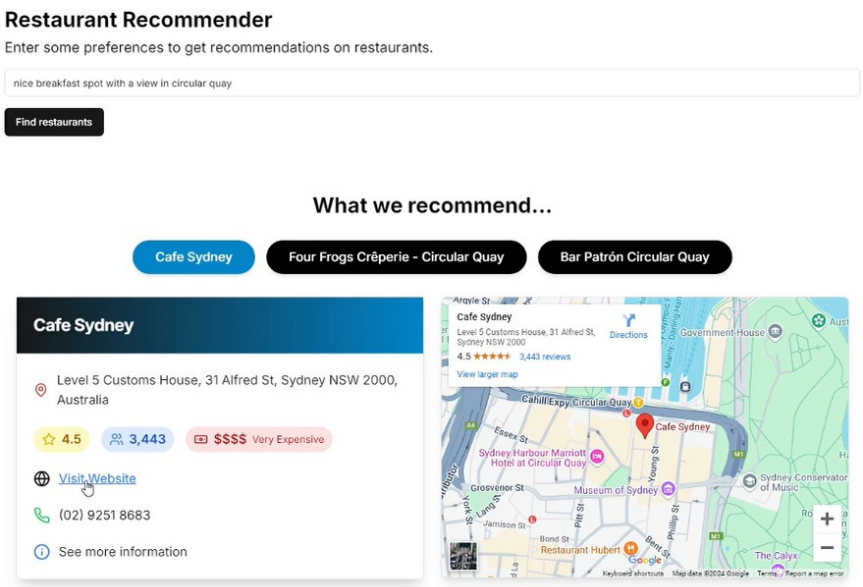
(b) Redirected Third Party Map from Restaurant Recommender

Figure 5. Map Functionalities of Restaurant Recommender.

Figure 6a,b below illustrate the web application's responses when user preferences are modified. By adjusting criteria such as budget constraints and dining preferences, the system generates distinct recommendations, showcasing its ability to adapt dynamically to varying user inputs. This demonstrates the effectiveness of the proposed framework and the robustness of its design principles in delivering personalized and context-aware recommendations.



(a) Response Sample 1 of Restaurant Recommender



(b) Response Sample 2 of Restaurant Recommender

Figure 6. Responses of Web Application of Restaurant Recommender.

 doi:10.20944/preprints202502.1821.v1

1.2. Ethical and Privacy Consideration

To ensure that ethical AI principles and user privacy are followed, careful design considerations are adopted as shown in Figure 7 below. These ethical and privacy considerations ensure that the recommendation system is secure, transparent, and reliable, aligning with best practices in the deployment of responsible AI.

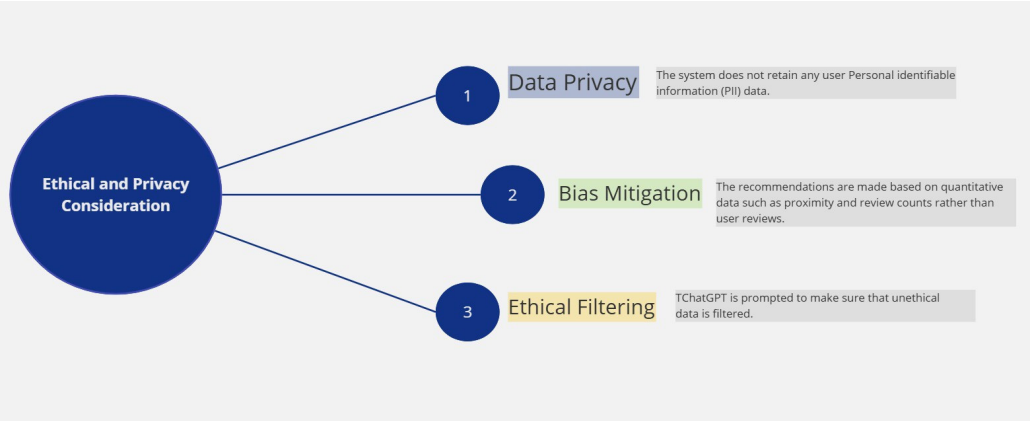


Figure 7. Ethical and Privacy Consideration.

Experimentation and Evaluation

The primary objective of this experiment is to evaluate the application of LLMs (specifically ChatGPT4o) in enhancing existing recommendation systems, such as Google, for restaurant recommendations. To evaluate the performance improvement by integrating LLMs with the existing recommendation systems a set of metrics has been defined. These include the management of complex queries, contextual accuracy, user satisfaction, and system response time, as presented in Table 2.

Management of complex queries demonstrated the ability of the system to process and interpret complex queries. It also reflects the system's capacity to understand multifaceted user requests. This criteria was measured by the system's ability to correctly interpret and apply specified constraints while maintaining the accuracy and relevance of recommendations. The pass rate for basic queries was 88%, whereas complex queries achieved an 84% success rate, indicating that while the system effectively managed standard requests, nuanced multicriteria filtering introduced additional challenges.

Table 2. Evaluation Criteria.

Metrics(s)	Criteria
Handling of Complex Queries	Recommendations on varied choice in a single query, such as "affordable" or "family friendly."
LocationBased Results	Recommendations based on the user's specified location i.e., suburb, city, or street.
User Satisfaction	User satisfaction should be between the scale of 4.0 5.0
System Response Time	Response time of under 3.0 seconds.

Contextual (location based) results ensure that recommendations align with the user's specified geographical constraints, such as suburb, city, or street level details. The system leveraged the Google Places API to retrieve real time restaurant data and applied further filtering to rank results based on location relevance. High accuracy of contextual recommendations indicates if the system effectively adhered to spatial constraints.

 doi:10.20944/preprints202502.1821.v1

User satisfaction assessment ensures that the system is user friendly and helps in the quantification of user requirement satisfaction. It was assessed through a structured survey where participants evaluated key aspects of the system, including recommendation relevance, personalization, handling of special requirements, and overall usability. The aspects included the system response time, food preference accuracy, overall recommendation quality, and customization capability.

System response time was measured across all test case user inquiries to determine whether the recommendation system could deliver results within the predefined 3.0second threshold. The slightly faster response time for complex queries can be attributed to the smaller dataset retrieved from the API. A small number of test cases exceeded the 3.0second threshold, particularly those involving high complexity requests, but these instances rep resented outliers rather than systemic inefficiencies.

For system experimentation, two sets of user requests were formulated. Each is for complex and simple requests, respectively. Successful query testing requires that each query satisfy all the criteria outlined in Table 2. Failure to meet any criterion resulted in an overall failure of the query. Figure 8 shows the failure and passing rate of queries.

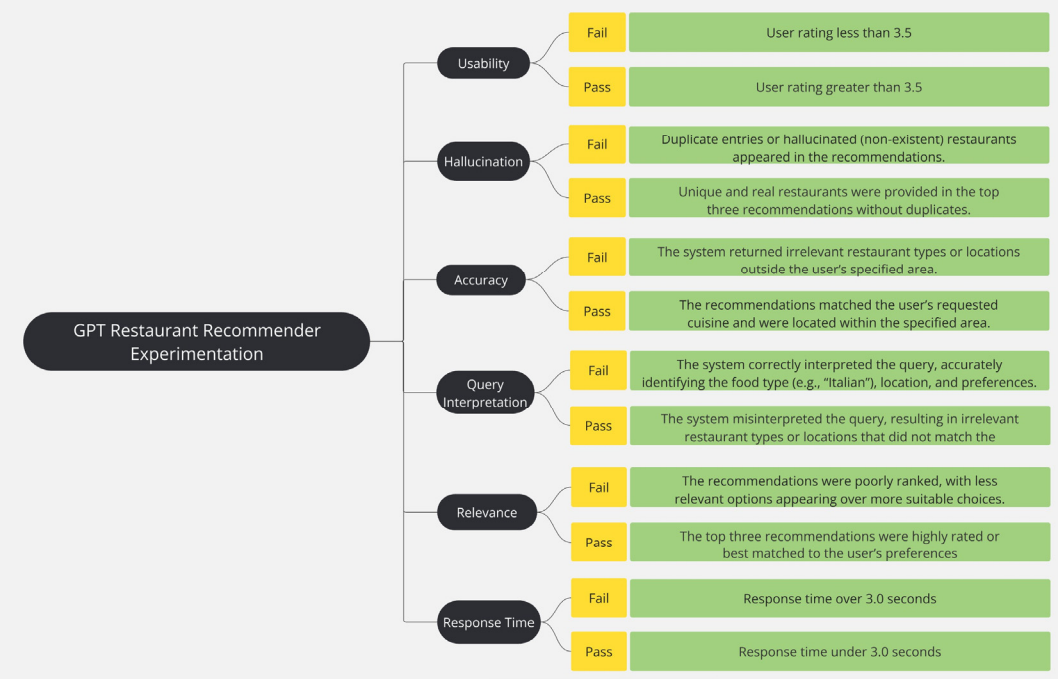


Figure 8. Evaluation Criteria of Fail and Pass Cases.

Evaluation Results and Discussion

This section discusses the findings of the experimental results against the set criteria as explained in Table 2. The system has been evaluated on 25 basic queries focusing on straightforward inquiries (e.g., Italian restaurant in Newtown or Sushi in Sydney CBD) and 25 complex queries which included additional descriptors such as affordable, vegetarian, family friendly, or ambiance preferences. These queries required the system to process nuanced input and provide recommendations based on both type and qualitative fac tors. Figure 9 shows the percentage of successful and failed queries.

From Figure 9, it is evident that the system demonstrated a high accuracy of 88% for basic queries and 84% for complex queries. The system's accuracy suggests that the system reliably interprets the main elements of the query. However, the slightly lower pass rate for complex queries indicates that multi criteria requests present an additional challenge. This finding aligns with [6], which notes the importance of prompt optimization for LLMs to handle spe cific API requests effectively. Their

 doi:10.20944/preprints202502.1821.v1

research shows that finetuned prompt designs improve response accuracy and speed, which could be a valuable approach for enhancing the handling of complex, multicriteria queries in this system.

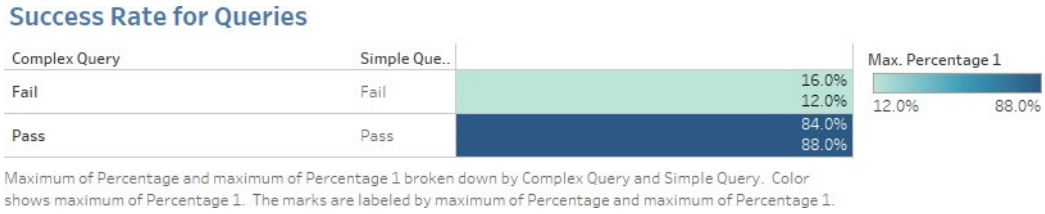


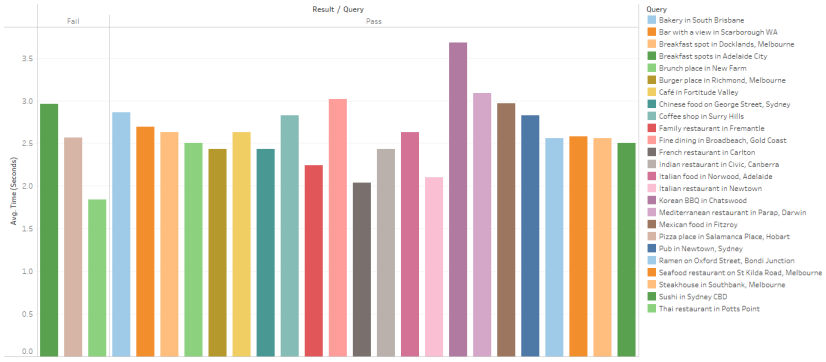
Figure 9. Queries Success Rate.

Figures 10a and 11a shows the queries along with their Pass/Fail status, while Figures 10b and 11b show the response time of the system for simple and complex queries respectively with their Pass/Fail status. From the results shown in Figures 10 and 11, it is evident that the system demonstrated efficient performance, achieving an average response time of approximately 2.57 seconds across both basic and complex queries. This response time aligns with expectations for real time applications, ensuring a smooth user experience with minimal delays. The consistency in response times across basic and complex queries highlights the system's robustness in managing both types of input. The slight decrease in response time for complex queries is noteworthy, as it suggests that the system's architecture is well optimized for handling additional parameters without a proportional increase in processing time. This capability is critical for maintaining user satisfaction, as faster responses enhance the overall experience.

The survey results indicate that users had a positive experience with the system, highlighting its usability and effectiveness. Participants evaluated various aspects, including food preference matching, personalization and customization of recommendations, overall search quality and relevance, system speed and response time, and how the recommendations compared to those from Google and other designed engines. Additionally, users rated their acceptance of the recommendations provided by the smart recommendation system. Figure 12 summarizes the average scores for each survey question.

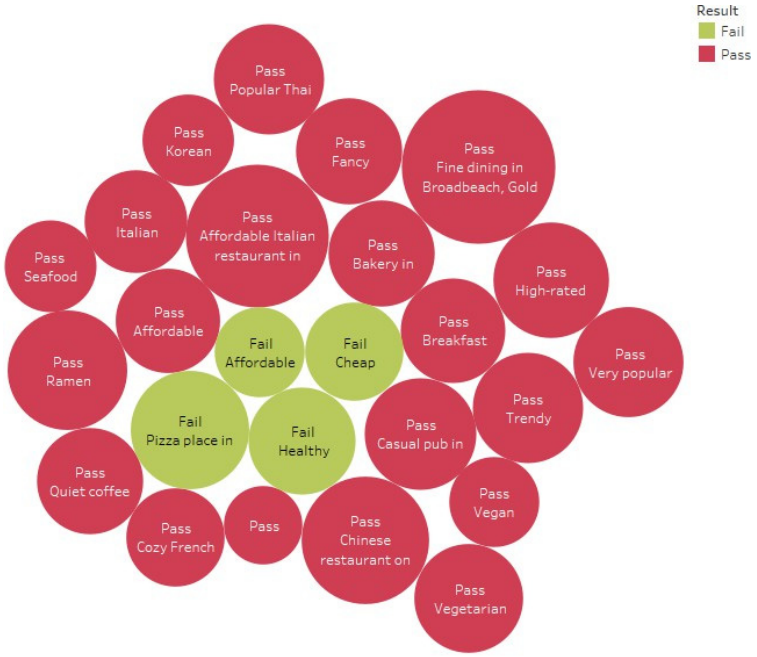
(a) Simple Query Results

 doi:10.20944/preprints202502.1821.v1



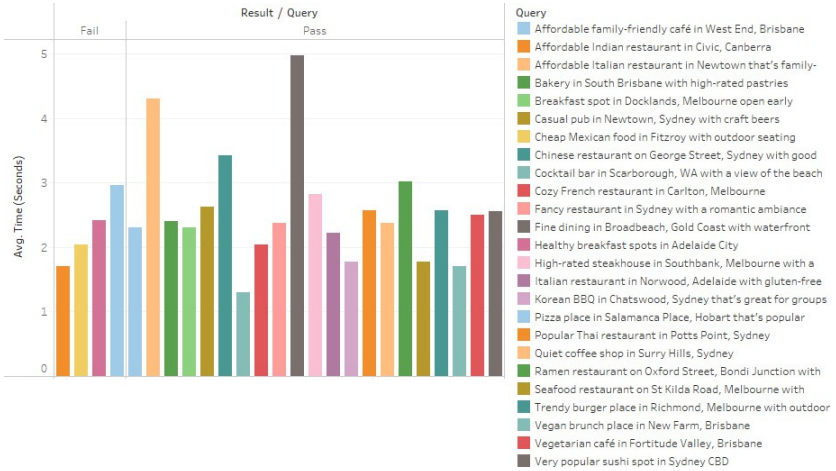
(b) Simple Query Status with Time

Figure 10. Simple Query Evaluation.



Result and Query. Color shows details about Result. Size shows average of Time (Seconds). The marks are labeled by Result and Query. The view is filtered on Result, which keeps Fail and Pass.

(a) Complex Query Results



(b) Complex Query Status with Time

 doi:10.20944/preprints202502.1821.v1

Figure 11. Complex Query Evaluation.

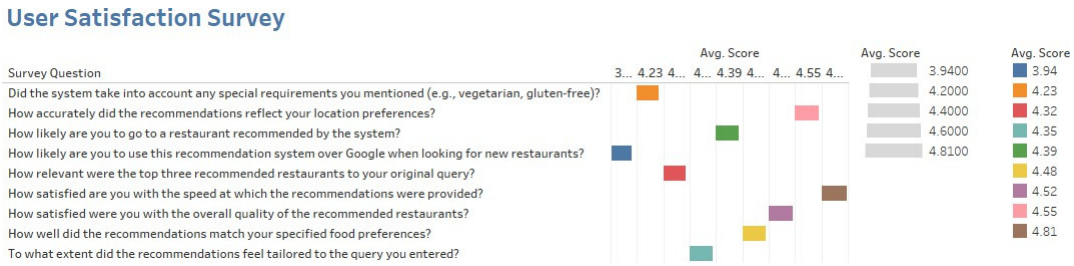


Figure 12. User Satisfaction Results.

However, users rated their likelihood of choosing this recommendation system over Google at a scale of 3.94/5.0. Although this score is slightly lower than other usability metrics, it suggests that users recognize the potential of the system as an alternative to Google, particularly if improvements are made in personalization and handling special requirements. The lower rating may also reflect the familiarity of users and the habitual reliance on Google's extensive database, highlighting an opportunity for further enhancement. Expanding data sources and improving personalization features could increase adoption and position the system as a competitive alternative.

Conclusion and Future Work

This research advances the interaction between LLMs and APIs to im prove recommendation systems, providing a more intuitive, responsive, and effective platform for interpreting complex user queries. The system achieves an accuracy of 84.88% (depending on the complexity of the query), with an average response time of 2.5 seconds and a high user satisfaction score of

4.39 out of 5.0. These results reinforce that the AI powered recommendation system effectively meets user needs in key areas, including food and location matching, speed, and overall recommendation quality. However, improving the handling of special requirements and further enhancing personalization could increase user satisfaction and strengthen the system's competitiveness with broader search engines such as Google. Despite its strong performance, a few limitations persist in the designed systems such as reliance on the Google Places API, which can introduce potential delays during peak system loads, impacting its response times. Moreover, few responses suffer from hallucinations. he findings suggest future research directions of a more sophisticated prompt design and an alternative source of real time data to replace the Google Places API. Moreover, this research provides a scalable framework for multidomain applications beyond restaurant recommendations in areas such as travel, entertainment, and healthcare, where real time AI-driven recommendations can offer substantial value. In conclusion, this work demonstrates the potential of integrating LLMs with APIs to build intelligent, real time recommendation systems, with continued advancements in personalization, scalability, and system optimization paving the way for broader adoption.

References

1. M. Johnsen, Developing AI Applications With Large Language Models, Maria Johnsen, 2025.
2. Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang, et al., Recommender systems in the era of large lan guage models (llms), IEEE Transactions on Knowledge and Data Engi neering (2024).
3. J. Li, J. Xu, S. Huang, Y. Chen, W. Li, J. Liu, Y. Lian, J. Pan, L. Ding, H. Zhou, et al., Large language model inference acceleration: A compre hensive hardware perspective, arXiv preprint arXiv:2410.04466 (2024).
4. A. Gokul, Llms and ai: Understanding its reach and impact, Preprints (May 2023). doi:10.20944/preprints202305.0195.v1. URL https://doi.org/10.20944/preprints202305.0195.v1

 doi:10.20944/preprints202502.1821.v1

5. L. Li, Y. Zhang, D. Liu, L. Chen, Large language models for generative recommendation: A survey and visionary discussions (2024). arXiv: 2309.01157. URL https://arxiv.org/abs/2309.01157
6. B. Silva, Y. G. Tesfagiorgis, (2023), Large language models as an inter face to interact with API tools in natural language, 2023.
7. S. G. Patil, T. Zhang, X. Wang, J. E. Gonzalez, (2023), Gorilla: Large language model connected with massive APIs. arXiv, 2023.
8. D. Luo, C. Zhang, Y. Zhang, H. Li, (2024), CrossTune: Blackbox few shot classification with label enhancement. arXiv.
9. K. I. Roumeliotis, N. D. Tselikas, D. K. Nasiopoulos, Precisiondriven product recommendation software: Unsupervised models, evaluated by gpt4 llm for enhanced recommender systems, Software 3 (1) (2024) 62–80.
10. D. Spinellis, (2024), Pair programming with generative AI, 2024.
11. J. Mao, D. Zou, L. Sheng, S. Liu, C. Gao, Y. Wang, (2024), Identify critical nodes in complex networks with large language models. arXiv, 2024.
12. J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, H. Zhang, Y. Liu, C. Wu, X. Li, C. Zhu, et al., How can recommender systems benefit from large language models: A survey, ACM Transactions on Information Systems (2023).
13. S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, (2024), MiniCPM: Unveiling the potential of small language models with scalable training strategies. arXiv, 2024.
14. N. Corecco, G. Piatti, L. A. Lanzendörfer, F. X. Fan, R. Wattenhofer, An llmbased recommender system environment, CoRR abs/2406.01631 (2024). URL https://doi.org/10.48550/arXiv.2406.01631
15. W. Fan, Z. Zhao, J. Li, Y. Liu, X. Mei, Y. Wang, J. Tang, Q. Li, Recommender systems in the era of large language models (llms), IEEE Transaction on Knowledge and Data Engineering (2023). doi: 10.48550/arXiv.2307.02046.
16. shadcn, shadcn/ui: Modern ui components for react, https://ui. shadcn.com/shadcn/ui Official Documentation, [Accessed 17012025] (2025).
17. Optimizing: Third Party Libraries — Next.js — nextjs.org, https://nextjs.org/docs/app/buildingyourapplication/ optimizing/thirdpartylibraries, [Accessed 17012025].
18. Vercel AI SDK, Openai — sdk documentation, https://sdk.vercel. ai/providers/aisdkproviders/openaiVercel AI SDK, [Accessed 17012025] (2025).
19. Next.js by Vercel The React Framework — nextjs.org, https:// nextjs.org, [Accessed 15012025].
20. Vercel: Build and deploy the best web experiences with the Frontend Cloud – Vercel — vercel.com, https://vercel.com/, [Accessed 1501 2025].