applied sciences

MDPI

Article

Multi-Layer Prompt Engineering for Intent Recognition in Travel Planning Assistants

Yijin Huang, Lanlan Ma and Yapeng Wang *

Faculty of Applied Sciences, Macao Polytechnic University, R. de Luís Gonzaga Gomes, Macao, China; p2315878@mpu.edu.mo (Y.H.); p2415988@mpu.edu.mo (L.M.)
* Correspondence: yapengwang@mpu.edu.mo

Abstract

Current travel planning tools suffer from information fragmentation, requiring users to switch between multiple apps for maps, weather, hotels, and other services, which creates a disjointed user experience. While Large Language Models (LLMs) show promise in addressing these challenges through unified interfaces, they still face issues with hallucinations and accurate intent recognition that require further research. To overcome these limitations, we propose a multi-layer prompt engineering framework for enhanced intent recognition that progressively guides the model to understand user needs while integrating real-time data APIs to verify content accuracy and reduce hallucinations. Our experimental results demonstrate significant improvements in intent recognition accuracy compared to traditional approaches. Based on this algorithm, we developed a Flask-based travel planning assistant application that provides users with a comprehensive one-stop service, effectively validating our method's practical applicability and superior performance in real-world scenarios.

Keywords: intent recognition; multi-layer prompt engineering; large language model; travel assistant

check for updates

Academic Editors: Grigorios Beligiannis and Paolo Renna

Received: 31 August 2025
Revised: 25 September 2025
Accepted: 23 October 2025
Published: 26 October 2025

Citation: Huang, Y.; Ma, L.; Wang, Y. Multi-Layer Prompt Engineering for Intent Recognition in Travel Planning Assistants. *Appl. Sci.* **2025**, *15*, 11442. https://doi.org/10.3390/app152111442

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of the global tourism industry, more and more tourists rely on online platforms for travel planning and decision-making [1]. The applications of smart travel assistants are becoming increasingly diverse, including itinerary recommendations, hotel reservations, attraction planning, transportation arrangements, etc. These assistants provide users with a more convenient and efficient travel experience by integrating technologies such as big data and artificial intelligence [2].

Currently, there are many tourism industry apps on the market, such as the recommendation systems of trip, Xiaohongshu, and other platforms [3]. These systems usually rely on historical user data and recommendation algorithms to provide travel suggestions, but there are the following problems: First, the recommendation algorithms of many platforms are mainly based on collaborative filtering or user portraits and lack a deep understanding of the user's specific intentions, resulting in generalized recommended content and difficulty meeting personalized needs. Second, in the travel planning process, users must search for information across multiple platforms, leading to fragmentation of information and increasing decision-making costs [4].

Due to the development of AI technology, it has been successfully applied in many different fields, such as image processing [5,6] and natural language tasks [7–10]. In recent

years, the application of Large Language Models(LLMs) (such as the Generative Pre-trained Transformer (GPT) series [11]) in dialog systems has made significant progress [12]. These models have powerful natural language understanding and generation capabilities, allowing intelligent assistants based on LLMs to interact with users more smoothly and provide more complex travel recommendations. However, LLMs still have the problem of "hallucination" in practical applications; that is, the model may generate information that is inconsistent with reality or completely fictitious. Therefore, how to improve the reliability of LLMs in intelligent travel assistants, reduce hallucination problems, and enhance intent recognition capabilities are important research directions in the current development of intelligent travel assistants.

The main contributions of this research can be divided into three categories:

- Proposing a multi-layer prompt engineering method: By introducing a hierarchical intent recognition strategy, the system's ability to understand user input is improved, and the hallucination phenomenon of LLMs is effectively reduced, thereby improving the accuracy of intent recognition.
- Conducting experimental verification and performance evaluation: Experimental tests were conducted on the CrossWOZ dataset [13]. The results show that compared with the basic model itself, the multi-layer prompt engineering approach proposed in this study significantly improves the accuracy of intent recognition and the overall recommendation effect.
- A mobile intelligent tourism case study was built to demonstrate the efficiency of our proposed method.

2. Related Work

Traditional dialog systems have evolved through three key stages. Initially, rule-based systems like ELIZA [14] used pattern matching but lacked flexibility and struggled with complex inputs. The second stage introduced machine learning methods such as Support Vector Machines (SVMs) [15] and Hidden Markov Models (HMMs) [16], which improved adaptability but required extensive labeled data and had limitations with complex intentions. The current deep learning era employs Recurrent Neural Networks (RNNs) [17], Long Short-Term Memory (LSTM) [18] networks, and Transformer [19] architectures (BERT [20,21], GPT [22]) that significantly enhance context understanding and response generation [23]. While these modern systems show substantial improvement in intent recognition and natural language generation, they still face challenges related to computational demands and data requirements [24].

In recent years, dialog systems powered by pre-trained LLMs like GPT [22], ChatGPT [25], and Claude have become mainstream [26]. These Transformer-based models generate coherent text and maintain contextual consistency across multiple dialog turns. Their advantages include superior context understanding without explicit state tracking, high-quality language generation from pre-trained knowledge, and strong generalization capabilities across various applications.

However, these systems face significant challenges. Hallucination remains a critical issue, with models sometimes generating false information [27]. Their intention control capabilities are limited by reliance on probability distributions, making it difficult to accurately identify complex user needs. Additionally, these models cannot directly access real-time information (weather, maps, flights), requiring integration with external APIs or knowledge bases. While modern LLM-driven dialog systems have significantly advanced language understanding and generation, they still require complementary technical approaches to improve stability, controllability, and user experience.

Intent recognition has evolved significantly within dialog systems, progressing from basic keyword matching methods (TF-IDF [28], BM25 [29]) to machine learning classifiers (SVMs [30], random forest), and finally to sophisticated deep learning approaches. Early keyword matching techniques, while simple to implement, struggled with semantic relationships and language variations. Machine learning methods improved generalization but remained limited by manual feature extraction. Modern deep learning approaches using RNNs, LSTMs, CNNs, and Transformer-based models like BERT and GPT have substantially advanced intent recognition by capturing complex semantic relationships, although challenges remain with long-tail intents and training data dependencies.

Prompt engineering has emerged as an effective alternative that leverages LLMs' inherent knowledge without additional training [31]. While existing research has explored various intent recognition methods, most approaches still face challenges with personalization and factual accuracy [32]. Our study bridges this gap by implementing a multi-layer prompt engineering approach that decomposes complex tasks into manageable subtasks with independent prompts, enabling step-by-step reasoning and reducing error rates. Unlike previous work that relies solely on a model's-internal knowledge, we integrate external APIs (maps, weather, etc.) to form a complementary system that combines the flexibility of prompt engineering with the factual reliability of real-time data. This hybrid approach addresses the hallucination problem that is common in LLMs while maintaining personalization capabilities, particularly for complex travel planning needs, where fixed label systems struggle. Our method represents a significant advancement over existing techniques by balancing the trade-off between personalization and factual accuracy in travel planning assistants, substantially improving intent recognition accuracy and overall user experience.

3. Proposed Method

Figure 1 illustrates the overall architecture of our proposed multi-layer intent recognition system for algorithms for improving the accuracy of identifying user intent in a travel planning assistant. The proposed method utilizes structured prompts and hierarchical query processing to extract relevant information, reduce hallucinations, and improve response quality.

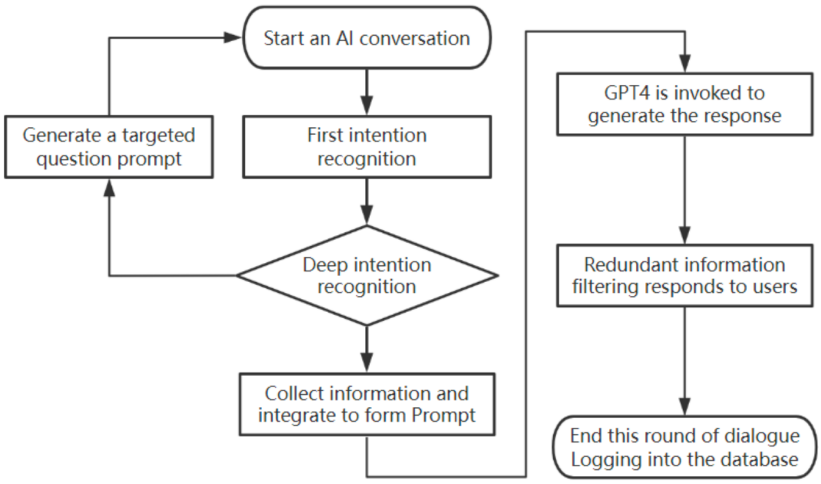


Figure 1. Workflow of multi-layer intent recognition.

The system consists of three key components: city extraction, intent classification, and service invocation. Each module runs sequentially, refining the user's input. The intent classification module acts as the central control flow, determining which final service (e.g.,

route generation, Point of Interest (POI) retrieval, or weather query) is executed in the final stage.

City Extraction: The first layer focuses on identifying the cities mentioned in the user's query. The structured prompt instructs the language model to extract city names based on predefined examples. The extracted city information is crucial for contextualization of subsequent intent classification Algorithm 1.

Algorithm 1 City Extraction using Language Model

Require: User query q
Ensure: Extracted city name $city$
Construct the query string $question$ by appending q to $prompt_city$
Invoke the language model llm with $question$ and retrieve output x
Initialize $city \leftarrow$ ""
Attempt to parse x as JSON
if parsing is successful **then then**
    Extract city name $city \leftarrow x[\text{"}city\text{"}]$
    Print extracted city information for debugging
else
    Print error message and raw output x
end if
return $city$

Intent Classification: Once the city is determined by the first layer, the system processes the user's query to determine the precise intent, which may include route planning, hotel recommendations, restaurant suggestions, or real-time weather requests. The language model is guided by structured examples to accurately classify this intent. This classification is crucial, as it acts as the system's central control flow, directing the request to the appropriate back-end service module for the final stage. If the intent is route planning, the system proceeds to the Route Recommendation module. Conversely, if the intent falls under a Real-time Query category (such as a request for hotels, restaurants, or weather), the system bypasses the route generation process and instead triggers the external API invocation module to retrieve structured, real-time data. In the case of API invocation, the city name extracted from the first layer is mandatorily used as a contextual filter, ensuring the accuracy and relevance of the retrieved POI data Algorithm 2.

Algorithm 2 Intent Recognition and Response Generation

Require: User query q, city name $city$
Ensure: Generated response $response$
Construct query string $question \leftarrow prompt_intention + q$
Invoke LLM with $question$, retrieve output x
Parse x as JSON, extract intent $intent \leftarrow x[\text{"}intent\text{"}]$
if intent is recognized **then then**
    Retrieve relevant results based on intent
    Format results into $result_simple$
    Generate response using template
else
    return Empty dictionary
end if
return $response$

Route Recommendation: This module is only activated if the preceding intent classification step confirms route planning intent. For travel-related queries, the system extracts key places and generates the best routes, while the algorithm retrieves the place names and makes travel plans accordingly Algorithm 3.

Algorithm 3 Get Travel Route

Require: User query q, City name $city$
Ensure: Route recommendation $result$
CConstruct query $question \leftarrow prompt_route + q +$ "\nOutput3:"
Invoke LLM with $question$, retrieve output x
Initialize $result \leftarrow$ ""
if x is valid JSON **then then**
    Extract places $places \leftarrow x[$"$places$"$]$
    for each place sequence p in $places$ **do do**
        Retrieve routes $route1 \leftarrow get_route_result(p[0], p[1], city)$
        Retrieve routes $route2 \leftarrow get_route_result(p[1], p[2], city)$
        Summarize $route1$, $route2$ using LLM
        Append routes to $result$
    end for
else
    Handle parsing error, log raw output
end if
return $result$

The proposed multi-layered hinting approach effectively improves the accuracy of intent recognition by building hints at different stages. This approach minimizes hallucinations and enhances contextual awareness, making it suitable for AI-driven travel planning applications. Future work will explore optimization techniques to further improve accuracy and efficiency.

4. Experiments

The experiments are designed to verify the effectiveness of the proposed multi-level intent recognition algorithm in improving accuracy and reducing hallucination when applied to LLMs in the travel domain. By combining carefully prepared datasets with controlled experimental settings, we aim to explore both the baseline capabilities of different models and the improvements achieved through our optimization method.

4.1. Dataset and Preprocessing

The dataset used in this study is the CrossWOZ dataset, which is a large task-oriented Chinese conversation dataset. The dataset covers five domains—hotels, restaurants, tourist attractions, subways, and cabs—and contains more than 6000 conversations and 100,000 dialog turns. Since this dataset contains user intent, dialog states, and relevant travel information, it is suitable for evaluating intent recognition models, especially for the travel domain.

During data processing, the data is first loaded through Python 3.10's json module. The code reads the data from the JSON file and loads it into the dataset. To ensure diversity in the sample, the code first selects travel-related conversations from the dataset. The system extracts the intent of the user based on the keywords in the conversation and further checks whether the conversation contains keywords such as "attractions", "hotels", or "restaurants".

When sampling, 100 conversations are selected by default. If the number of data points is less than 100, all eligible data points will be returned to avoid oversampling. Data sampling is realized through the random.sample() method, which ensures the randomness and representativeness of sampling. Finally, the sampled data is saved as a new JSON file for subsequent analysis and use.

4.2. Experimental Design

This experiment applied the multi-level intent recognition algorithm to four models (GPT-3.5, GPT-4, Qwen2-7B, Llama3-8B) to evaluate the performance difference before and after optimization. We first loaded data from the data source and preprocessed it to ensure that the data is compatible with subsequent model inputs. Then the four original models were used to predict the preprocessed data, and the output results of each model were recorded to obtain preliminary experimental results. Next, the designed multi-level intent recognition algorithm was applied to optimize the original model to improve the accuracy of intent recognition and enhance model performance. In the experimental group, the optimized model was used to experiment on the same dataset, the output results were recorded, and the changes in answers before and after optimization were compared. Finally, the accuracy and hallucination rate were used as evaluation indicators to analyze the performance differences between the original model and the optimized model.

In order to verify the effect of multi-level prompt word algorithm on improving the accuracy of intent recognition, this study designed two experimental settings: the experimental group and the control group. The difference between the experimental group and the control group is whether the multi-level prompt word algorithm is used for optimization, as follows:

Control Group: The selected model is directly used for intent recognition without applying the multi-level prompting algorithm. In this setting, the model directly generates responses based on user input and records the generated responses as predicted labels. This set of experiments aims to evaluate the baseline performance of the model without any optimization.

Experimental Group: In this setting, the model processes user input through a multi-level prompting algorithm before performing intent recognition. This preprocessing helps the model better understand user intent, reduce hallucinations, and improve accuracy, allowing the impact of the algorithm on intent recognition to be evaluated.

Evaluation Metrics and Calculation Methods

First, during the experiment, we recorded the user input, expected answer, and actual answer of each dialog to determine whether the model's prediction was correct (i.e., whether the model output matched the actual answer). Based on these records, we further calculated the accuracy to measure the overall performance of the model. Accuracy is used to measure the proportion of correct predictions, and its definition is shown in Formula (1).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{1}$$

In order to further explore the authenticity and reliability of the content generated by the large model in tourism planning tasks, this study introduced a second evaluation indicator in addition to the traditional accuracy evaluation—the hallucination rate (HR). This indicator aims to quantify the proportion of fictitious information in the model's answers, thereby more comprehensively evaluating the output quality of the model.

The hallucination rate (HR) is based on the following types of hallucination definitions:

1. Fictional Places: Responses include attractions or restaurants that do not exist in the real world.
2. False Directions: Responses describe routes or geographic details between locations that are clearly wrong.
3. Fictional Information: Responses describe background or historical information that does not match the actual information.

4. Irrelevant Content: Additional information that is irrelevant to the user's query and may cause confusion is included.
5. Misattributed Content: Responses describe features or attractions in one city that are incorrectly attributed to another city.

During the experiment, we recorded the user input, expected answer, and actual model output and manually determined whether each answer had hallucinations based on the above criteria. The hallucination rate (HR) was calculated using Formula (2).

$$\text{HR} = \frac{\text{Number of responses containing hallucinations}}{\text{Total number of responses}} \quad (2)$$

5. Results and Discussion

This experiment set the accuracy and hallucination rate as evaluation indicators to measure the intent recognition performance of the model. The improvement effect of the algorithm on the model can be demonstrated in multiple dimensions. Tables 1 and 2 compare the accuracy and hallucination rate before and after applying our method, with the percentage improvements highlighted.

Table 1. Model performance comparison.

Model	Accuracy	Accuracy †	Improvement
Qwen2-7B	76%	84%	8%
Llama3-8B	72%	87%	15%
GPT-3.5	87%	**95%**	8%
GPT-4	89%	**95%**	6%

† Means accuracy obtained using the proposed method. Bold indicates the best result among all results.

Table 2. Model hallucination rate comparison.

Model	Hallucination Rate	Hallucination Rate †	Improvement
Qwen2-7B	22%	3%	↓ 19%
Llama3-8B	18%	6%	↓ 12%
GPT-3.5	5%	0%	↓ 5%
GPT-4	2%	0%	↓ 2%

† Means hallucination rate with the proposed method.

As can be seen from the tabular results, all models benefited from the multi-level intent recognition optimization, with both accuracy and phantom rate being improved. Notably, Qwen2-7B and Llama3-8B showed significant improvements, with their accuracy increasing by 8% and 15%, respectively, and their phantom rate being significantly reduced. This shows that the approach is particularly effective for small and medium-sized models.

In contrast, the improvements for GPT-3.5 and GPT-4, which already showed high baseline performance, were more modest. After optimization, the phantom rates of both GPT-3.5 and GPT-4 dropped to zero, indicating that the integration of external APIs and accurate intent recognition can further stabilize the output of LLMs, even if their initial phantom rates are already low. Overall, the experimental results verify the effectiveness of the proposed multi-layer approach in improving the reliability and factual correctness of model responses in travel planning tasks.

The experimental results demonstrate that the proposed multi-level intent recognition algorithm significantly enhances dialog understanding by improving accuracy and reducing hallucination rates across models. Lightweight models like Qwen2-7B and Llama3-8B see the greatest benefits, with notable accuracy gains (8% and 15%) and reductions in hallucination rates (from 22% to 3% and 18% to 6%). In contrast, larger models like GPT-3.5

and GPT-4, which already perform well, experience smaller but meaningful improvements, including a reduction of hallucination rates to zero. However, small models still face residual challenges in complex scenarios due to limited reasoning capabilities. These findings highlight the practical value of structured prompt optimization, which is especially effective for cost-efficient, smaller models, offering promising applications in tasks like travel planning.

6. Use Case

We created a travel assistant to better demonstrate our proposed method. The main design of the recommendation dialog system is shown in Figure 2.

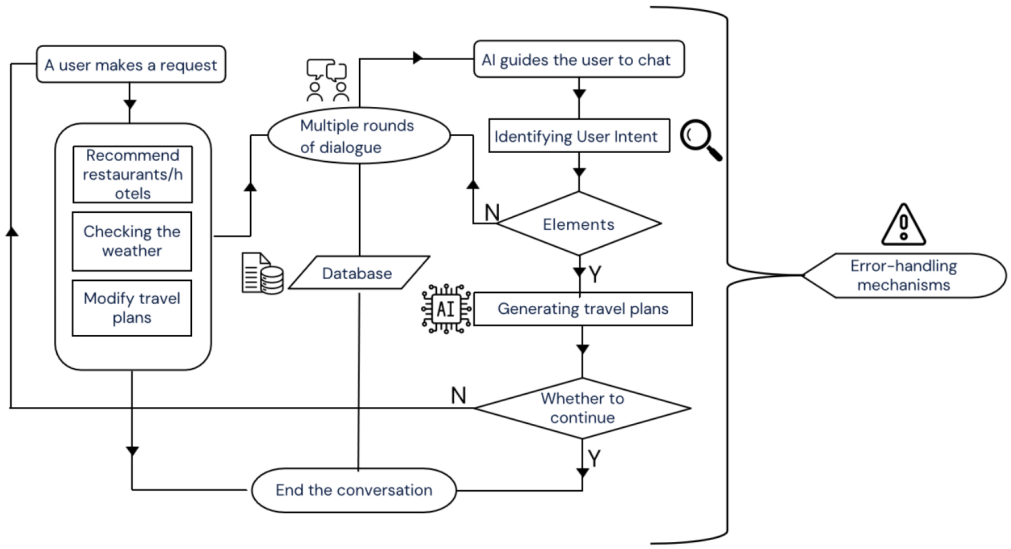


Figure 2. Architecture of dialog system.

6.1. System Function Introduction

We built a web application focusing on personalized travel itineraries, aiming to provide users with comprehensive travel planning and information services. The following is a brief introduction to the main functions of the application:

- Conversational chat: Users can express their travel needs and preferences to the smart assistant through natural language conversations, and the smart assistant will provide personalized travel suggestions and information based on the user's input.
- Travel itinerary generation: The system can automatically generate a detailed travel itinerary based on the number of days and destinations specified by the user, and users can obtain daily scenic spot recommendations and activity arrangements.
- Personalized recommendations: The system supports intent-driven, dedicated information retrieval for crucial POIs, including suitable restaurants and hotels, in addition to scenic spots. This functionality is implemented via our multi-layer intent recognition, which directs the request to external APIs for real-time data.
- Scenic spot navigation: The system provides navigation services between scenic spots to help users plan the best travel routes and ensure the smooth progress of their travel.
- Weather query: Users can query real-time weather information for the destination to better arrange travel plans.
- Error handling: The entire system is equipped with multiple error handling mechanisms to help developers and users quickly locate error components.
- Conversation history data storage: With a database to save relevant session records, it is easy to check at any time.

6.2. Implementation Details

This section introduces the details for building the demo system.

6.2.1. Main Function

The front-end uses HTML+CSS to build the layout and Flexbox to achieve responsive design and dynamically generates navigation links through Flask to ensure that page links are displayed correctly in different environments.

We use Bootstrap 5 to implement responsive layout, and JavaScript handles form submission and background data interaction. After obtaining weather data through APIs, we update the page display, including temperature, humidity, wind speed, and other information.

The code implements a travel assistant system based on the Flask framework, providing personalized travel planning services, including user interaction, weather query, tourist attraction recommendations, etc. The demo system uses the OpenAI GPT-4 model for intent recognition and integrates a database to record user input and system responses for later management. External APIs, such as weather queries and Baidu Map APIs, provide real-time weather, attraction, and dedicated POI information for hotels and restaurants.

6.2.2. Dialog Data Storage

The system uses the SQLite database to store chat records, which includes two tables: a conversation table and a message table. The conversation table records the start time and title of each conversation, and the message table stores the detailed information of each message, including conversation ID, role, content, timestamp, etc. This design meets the basic needs of chat record storage and is also convenient for subsequent queries and management.

SQLite was chosen as the database because it is lightweight and does not require a separate server, making it ideal for small projects and local applications. SQLite stores data in a single file, making it easy to deploy and maintain. In addition, it supports standard SQL syntax and can efficiently handle CRUD (create, read, update, delete) operations, making it ideal for storing structured data such as chat records.

6.2.3. API Information

This study involves the use of multiple APIs, including different LLMs used in the experimental phase and APIs for map and weather information used in the construction of the demo system. Table 3 lists all the API information used in this study.

Table 3. LLM and API information.

Use as	Name
LLM	GPT-4
LLM	Qwen2-7B
LLM	Llama3-8B
LLM	GPT-3.5
MAP	Baidu MAP API
Weather	Baidu MAP API

6.2.4. Demo

The welcome page is designed to provide an intuitive and attractive initial experience, as shown in Figure 3. It adopts a simple and elegant layout, with a text introduction on the left and a travel-themed picture on the right to enhance the sense of immersion. The title "Start your travel journey" complements the introduction and allows users to quickly

understand the core functions. The "Start Planning" button uses high-contrast colors to guide users to interact. The overall color scheme is mainly blue and white, conveying an atmosphere of free exploration.

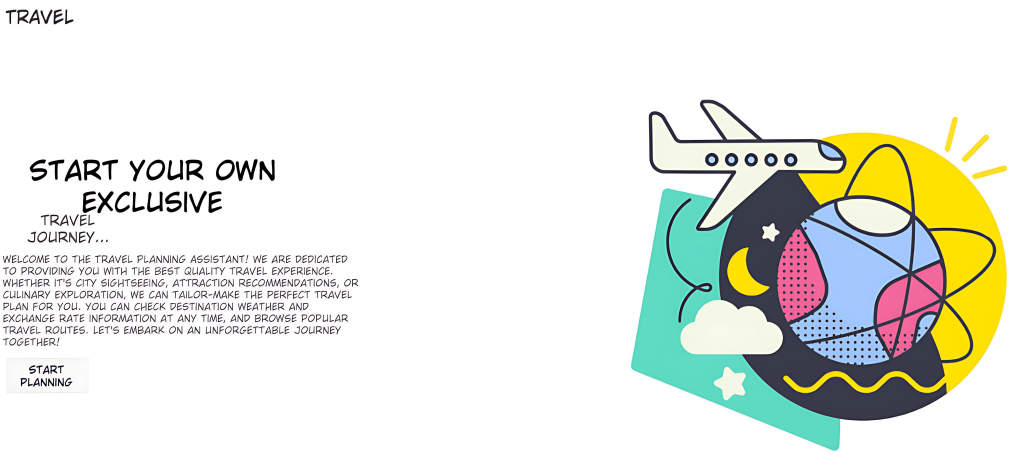
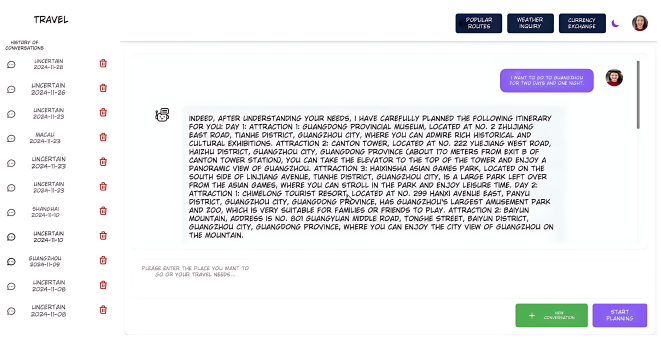
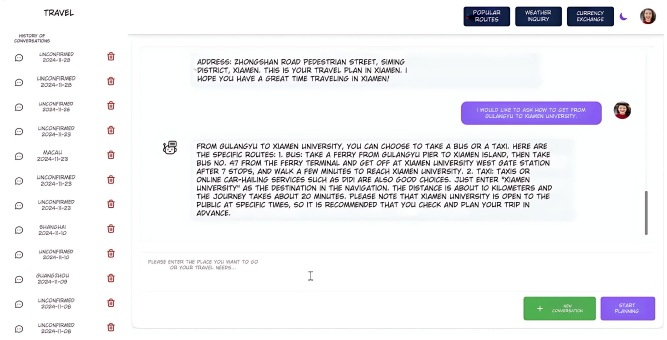


Figure 3. Welcome page.

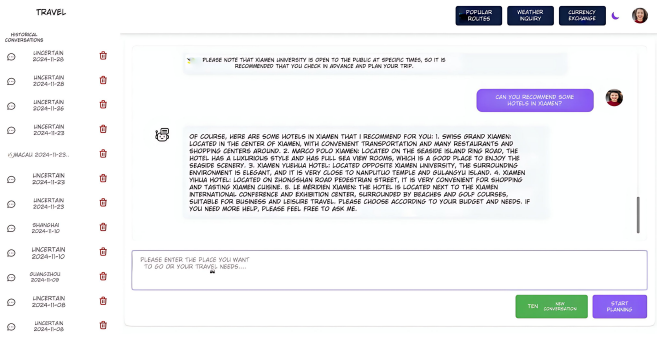
Figure 4 presents the conversation interface of the proposed system, which is composed of several functional modules. Specifically, the sidebar is used to display conversation records, allowing users to view and delete past chat histories. The title bar provides multiple functional buttons, including "Popular Routes," "Weather Query," and "Currency Exchange," as well as theme-switching options (dark and light mode). To further demonstrate the versatility of the system, four representative application scenarios are illustrated in Figure 4a–d, corresponding to travel planning, route planning, hotel recommendations, and restaurant recommendations, respectively.



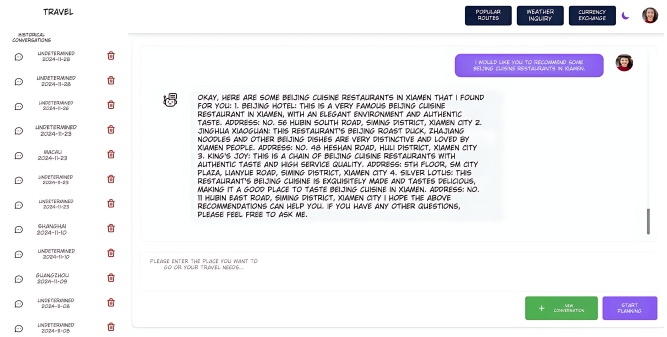
(**a**) Travel Plan



(**b**) Travel Route Plan



(**c**) Hotel Recommendation



(**d**) Restaurant Recommendation

Figure 4. Overview of the chat-based recommendation system.

7. Conclusions and Future Directions

This study presents a travel planning assistant system based on multi-layer prompt engineering that significantly enhances intent recognition accuracy while reducing hallucinations. By implementing a hierarchical approach to prompt engineering combined with external APIs (weather, maps, etc.), the system demonstrated substantial improvements in intent recognition on the CrossWOZ dataset. The Flask-based framework offers excellent scalability for future optimizations. Our contributions include a novel multi-layer prompt engineering method that improves task understanding, integration with practical external services creating a highly applicable system, and experimental validation showing significant performance gains over traditional single-layer prompting techniques. This research not only advances natural language processing applications in smart tourism but also provides valuable insights for further exploration of multi-layer prompt engineering.

Our future research will focus on optimizing prompt design with dynamic adjustment strategies, incorporating deep learning techniques like reinforcement learning, expanding data sources to improve recommendation credibility, developing multi-modal interaction capabilities, and conducting large-scale testing in real-world scenarios. These improvements aim to enhance the intelligent travel assistant system's adaptability, personalization, and overall user experience. In addition, future work will also extend the system to support more languages, enabling broader applicability across different linguistic and cultural contexts.

Author Contributions: Conceptualization, Y.H.; Methodology, Y.W.; Software, Y.H.; Validation, L.M.; Formal analysis, Y.H.; Investigation, Y.H. and L.M.; Resources, L.M.; Data curation, Y.H.; Writing—original draft, Y.H.; Writing—review and editing, L.M. and Y.W.; Visualization, Y.H.; Supervision, Y.W.; Project administration, Y.W.; Funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Macao Polytechnic University under grant number RP/FCA-03/2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are derived from the publicly available open-source dataset CrossWOZ (https://github.com/thu-coai/CrossWOZ (accessed on 22 October 2025)). This dataset is freely accessible and can be downloaded without restriction. All analyses in this study were conducted using the original version of the dataset.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ning, Y.M.; Hu, C. Influence mechanism of social support of online travel platform on customer citizenship behavior. *Front. Psychol.* **2022**, *13*, 842138. [CrossRef] [PubMed]
2. Wang, Y.; Lam, C.T.; Choi, K.C.; Ng, B.; Siu, K.M.; Lei, P.; Yang, X. An Open Tourism Mobile Platform with Personalized Route Planning and Voice Recognition. In Proceedings of the ICIT '21: 2021 9th International Conference on Information Technology: IoT and Smart City, Guangzhou, China, 22–25 December 2021; pp. 337–345. [CrossRef]
3. Li, Y.; Lee, F.L. Social media and the negotiation of travel photo authenticity in the making of Wanghongjingdian in China. *Tour. Stud.* **2024**, *24*, 221–245. [CrossRef]
4. Alsahafi, R.; Mehmood, R.; Alqahtany, S. A Machine Learning-Based Analysis of Tourism Recommendation Systems: Holistic Parameter Discovery and Insights. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 1369. [CrossRef]
5. Tse, R.; Monti, L.; Im, M.; Mirri, S.; Pau, G.; Salomoni, P. DeepClass: Edge based class occupancy detection aided by deep learning and image cropping. In Proceedings of the Twelfth International Conference on Digital Image Processing (ICDIP 2020), Osaka, Japan, 19–22 May 2020; Volume 11519, pp. 20–27.

6. Beyan, E.V.P.; Rossy, A.G.C. A review of AI image generator: Influences, challenges, and future prospects for architectural field. *J. Artif. Intell. Archit.* **2023**, *2*, 53–65.
7. Zhou, M.; Duan, N.; Liu, S.; Shum, H.Y. Progress in neural NLP: Modeling, learning, and reasoning. *Engineering* **2020**, *6*, 275–290. [CrossRef]
8. Li, J.; Yang, Y.; Mao, C.; Pang, P.C.; Zhu, Q.; Xu, D.; Wang, Y. Revealing patient dissatisfaction with health care resource allocation in multiple dimensions using large language models and the international classification of diseases 11th revision: Aspect-based sentiment analysis. *J. Med Internet Res.* **2025**, *27*, e66344. [CrossRef] [PubMed]
9. Li, J.; Yang, Y.; Chen, R.; Zheng, D.; Pang, P.C.; Lam, C.K.; Wong, D.; Wang, Y. Identifying healthcare needs with patient experience reviews using ChatGPT. *PLoS ONE* **2025**, *20*, e0313442. [CrossRef] [PubMed]
10. Yang, Y.; Li, J.; Guo, J.; Cheong-Iao Pang, P.; Wang, Y.; Yang, X.; Im, S.K. Performance Evaluation and Application Potential of Small Large Language Models in Complex Sentiment Analysis Tasks. *IEEE Access* **2025**, *13*, 49007–49017. [CrossRef]
11. Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J.W.; Kreps, S.; et al. Release strategies and the social impacts of language models. *arXiv* **2019**, arXiv:1908.09203. [CrossRef]
12. Xie, J.; Zhang, K.; Chen, J.; Zhu, T.; Lou, R.; Tian, Y.; Xiao, Y.; Su, Y. Travelplanner: A benchmark for real-world planning with language agents4. *arXiv* **2024**, arXiv:2402.01622.
13. Zhu, Q.; Huang, K.; Zhang, Z.; Zhu, X.; Huang, M. Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 281–295. [CrossRef]
14. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine, 1966. *Commun. ACM* **1966**, *9*, 36–45. [CrossRef]
15. Andrews, P.; Manandhar, S. A SVM Cascade for Agreement/Disagreement Classification. In *Traitement Automatique des Langues, Volume 50, Numéro 3: Apprentissage automatique pour le TAL [Machine Learning for NLP]*; ATALA (Association pour le Traitement Automatique des Langues): Montpellier, France, 2009; pp. 89–107.
16. Li, Z.; Wang, Z.; Sun, X. Research Applications of Hidden Markov Models in Speech Recognition. In Proceedings of the 2023 International Conference on Advances in Artificial Intelligence and Applications, Wuhan, China, 18–20 November 2023.
17. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
18. N, Y.B.; J, S.L.; S, R.; Tiwari, M.; K, A. Improved tweets in english text classification by lstm neural network. In Proceedings of the 2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI), Tiruchengode, India, 19–20 October 2023.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
20. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
21. Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X. Tinybert: Distilling bert for natural language understanding. *arXiv* **2020**, arXiv:2004.04726. [CrossRef]
22. Singh, P.; Jain, B.; Sinha, K. Evaluating bert and gpt-2 models for personalised linkedin post recommendation. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 6–8 July 2023; pp. 1–7.
23. Ho, N.L.; Lee, R.K.W.; Lim, K.H. Btrec: Bert-based trajectory recommendation for personalized tours. *arXiv* **2023**, arXiv:2310.19886 [CrossRef]
24. Kumar, H.; Kumar, G.; Singh, S.; Paul, S. Text summarization of articles using lstm and attention-based lstm. In *Machine Learning and Autonomous Systems, Proceedings of the ICMLAS 2021, Tamil Nadu, India, 24–25 September 2021*; Springer: Singapore, 2023.
25. Gao, L.; la Tour, T.D.; Tillman, H.; Goh, G.; Troll, R.; Radford, A.; Sutskever, I.; Leike, J.; Wu, J. Scaling and evaluating sparse autoencoders. *arXiv* **2024**, arXiv:2406.04093. [CrossRef]
26. Scheiner, J.; Williams, I.; Aleksic, P. Voice search language model adaptation using contextual information. In Proceedings of the 2016 IEEE Spoken Language Technology Workshop (SLT), San Diego, CA, USA, 13–16 December 2016; pp. 253–257.
27. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 42. [CrossRef]
28. Jain, S.; Jain, S.K.; Vasal, S. An effective tf-idf model to improve the text classification performance. In Proceedings of the 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT), Jabalpur, India, 6–7 April 2024; pp. 1–4.
29. Kadhim, A.I. Term weighting for feature extraction on twitter: A comparison between bm25 and tf-idf. In Proceedings of the 2019 International Conference on Advanced Science and Engineering (ICOASE), Zakho-Duhok, Iraq, 2–4 April 2019; pp. 124–128.

30. Kazemi, A.; Boostani, R.; Odeh, M.; AL-Mousa, M.R. Two-layer svm, towards deep statistical learning. In Proceedings of the 2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI), Zarqa, Jordan, 29 November–1 December 2022; pp. 1–6.
31. Chaubey, H.K.; Tripathi, G.; Ranjan, R.; Gopalaiyengar, S.k. Comparative Analysis of RAG, Fine-Tuning, and Prompt Engineering in Chatbot Development. In Proceedings of the 2024 International Conference on Future Technologies for Smart Society (ICFTSS), Kuala Lumpur, Malaysia, 7–8 August 2024; pp. 169–172. [CrossRef]
32. Yong, E.D.; De Jesus, L.C.M.; Abordo, M.J.R.; Co, M.J.; Derige, D.F.; Perez, D.J.M. Chatbot and voicebot feature for ramibot using deep learning sequential model for intent classification. In Proceedings of the 2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Coron, Philippines, 19–23 November 2023; pp. 1–5.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.