

CSCE420 Final AI Project

Adi Ravindra - Luke VanWagner - Brayton Lee

Abstract

Trip planning applications have a severe deficiency when it comes to dining. Once at the destination, everything from the weather, to moods and experiences, to simply how tired the party is may influence what type of food is desired. This project sought to correct this problem. After a short conversation with an AI assistant discussing needs and desires, meal suggestions will be provided for the trip, which can be updated during the trip at any time. Though some suggestions are limited by human nature or lags in real-time updates from information aggregators, this application provides an excellent option for planning for the future, while taking in to account human group dynamics and the ephemeral need of human needs and emotions.

Introduction

Planning a trip to a new destination can be difficult and time-consuming. Travelers are bombarded with advice from well-meaning friends, family, and countless websites. They must then sort through an endless list of flights and hotels, and find the most exciting things to do. Many people end up feeling overwhelmed, or participate in activities that do not match their interests. Expecting travelers to know what their party might want to eat in the planning phase of a trip might be impossible. Further, families with children or special dietary needs require a knowledgeable guide that may not be available. [2] With a cultivated list of select restaurants based on needs and a description of what is desired at the moment, our goal is to alleviate this process by condensing it from hours to minutes, while giving the user a concise and satisfactory experience.

Related Work

A list of referenced and inspirational projects is below. They are all included because each solves a problem in an interesting or unique manner. All are missing a useful method for discovering and selecting appropriate meals beyond inserting meal selection times into the itinerary.

- mindtrip.ai [13] (mindtrip.ai) - A chat-based platform where the user describes a destination and parameters for a desired trip. The user can also request specific needs and requirements for the trip such as number of rooms or specific needs. It can also make suggestions for flights and hotels. The food selection is limited to entering types of cuisine and no actual food selection.

- usevacay.com [10] (usevacay.com) - Users select a range of dates and needs for a location or region, and are provided with a list of available activities that fit within the limitations that are given. Selections are made by clicking through predetermined categories that will assist the user in reducing the options presented.
- tripplanner.ai [11] (tripplanner.ai) - A chat-based platform that uses a back-and-forth series of friendly and conversational questions to determine the optimal vacation for the traveler and their party. This tool is limited to only a selection of activities and no restaurants or other places to eat.
- An evaluation of TravelAgent [3] (<https://arxiv.org/abs/2409.08069>) - A study by Cornell evaluating performance of a trip planner site found that many AI agents and models are severely limited by their understanding and use of dynamic real-time data. Most agents that are utilized do not have access to these real-time data, and this can severely limit their ability to provide accurate and useful suggestions to user questions.

Methodology

The trip planner will take in a variety of different constraints, including location, number of travelers, length of stay, and much more. It will then take all of these factors to find the optimal dining experience for the described parameters and the current phase of the trip. The expected tools and APIs for this planner are as follows:

Tech Stack -

- Python
- Tkinter
- TripAdvisor
- Google AI Studio

In the Tripadvisor API use 2 major APIs -

- Nearby Search - Given a location search, get up to 10 places found near given latitude/longitude
- Location Reviews - Using the Places ID, get up to 5 reviews (most recent) for the location

Basic Structure -

1. Collect information about user travel preferences and needs. Either from an input or through the selection of constraints
2. NLP Pipeline
 1. Tokenize the text extracted from user or from given inputs
 2. Determine the intent and extract key information –

3. Itinerary Construction
 1. Construct itinerary from information extracted from user
 2. Chain run LangChain to optimize itinerary
 3. Track information to fetch reviews from API,
 1. Construct query to send to API
4. API request/processing
 1. Send query in required format
 2. Receive JSON information and extract relevant information.
5. Result Handling
 1. Organize results based on user preferences
 2. Change output if needed based on user input
6. Show Output.

Results

Proffered results are excellent and well-selected at the end of each iteration. Even with many constraints, the selection would only occasionally have improper cuisine suggestions. The model may breakdown in trying to find appropriate choices when parsing through remote locations. This might be because of two different reasons. First, it may not recognize the geographical region that it is selecting from while preparing an API call, or it may reject returned results from an API call for being outside of the constraint location. Hallucinations are few and due to poor rationale for restaurant selection given its constraints. In the end, this product shows remarkable promise and a robust ability to work through incomplete direction and give a valid and viable solution for the user.

Discussion

Limitations

No product is without its limitations. In this case, several have been identified, though many are fundamentally at the mercy of the data that is garnered through the APIs. Other problems are influenced by the nature of human preferences.

Most critically, the source of the data may fall prey to a hivemind or brigading. At the heart of the data collection are people's reviews of products, little of which is controlled in any manner whatsoever [1]. Any individual can go to Tripadvisor or Google and review a product. Good, bad, or indifferent, unless there is a demonstrable inaccuracy, these opinions are just that, opinions. Further, the information given by users may be inaccurate, leading to conflicting information in the dataset that must be parsed or disregarded, a non-trivial challenge for even human-level "intelligence". A specific restaurant that is seen as "good" by the public may only

eventually be considered good because of the circular reinforcement of reviews stating that it is good. This is a known problem for data analysts.

Congruent to the hive mind effect is public brigading of reviews. Generally influenced by social media, groups of people who have never been serviced by the company in question may anonymously post a review [3]. These reviews generally tend to be negative because of a perceived problem that stems from the actions of an employee of the business or an event that took place there; however, occasionally they may be positive. Events like this, when noticed, can be controlled by the company collecting the reviews if it is noticed and if they care enough to make the correction.

For both the hivemind and brigading, controlling at the application level is difficult. It must be recognized that the customer is fundamentally responsible for recognizing and filtering these potential issues with the liberal application of common sense.

Additionally, incomplete, incorrect, or missing data can steer users toward making bad decisions. There are several methods that aggregators use to update their information: inertia, owner information, and scraping. Inertia is where user reports indicate the need for a review of the business in question. If enough user reports indicate a problem the algorithm will flag the account for review and a person will look into it, usually a volunteer [7]. Scraping is where website information is analyzed and used to populate business pages. All of these methods lag behind the real-time truth about the state of businesses. When referencing the aggregators through the use of their APIs, this stale information is mixed in with the good information. This presents two key problems. First, the user is offered something that is factually incorrect. Second, this stale information will hold more weight if reviews are stable in their opinions about the location. For example, a well-known barbeque restaurant decides to offer a specific menu for patrons with various dietary restrictions. It will take a long time for new reviews to supersede the old non-inclusive menu. This may lead the application to never recommend the restaurant to people who seek the new style of menu, reinforcing as a negative feedback loop.

There is also the challenge of enabling LLMs to be able to correctly interpret and return information for people based on soft language based on feelings. While “guessing” the next word is easy, providing relevant and correctly interpreted feedback is less easy [5]. A person might not know exactly how to communicate their desires and feelings to the application and get frustrated that they are not getting a reasonable response. Similar in the way that a parent might feel when trying to communicate to a toddler, there is a fundamental gap in the understanding of their wants and desires in the moment.

Smaller and non-chain establishments are also inherently at a disadvantage for notice by the collected data. When on vacation it can be difficult to get patrons to visit unknown locations and they will default to safe or known chains. Unless recommended by word of mouth, local restaurants may simply go unnoticed and unreviewed, perpetuating the cycle of positive feedback that chains receive [4]. There might be a solution to working around this by adding

additional prompts to the LLM encouraging it to preferentially consider local stores over chains unless the users prompt precludes them.

Finally, this style of application is fundamentally at a disadvantage of not being able to refine its processes by not being able to take in to account users that only use the application once or a few times, each with a poor user experience. This means that the feedback for the “optimal experience” will be biased to people the fit the model’s decisions as it stands at the final release iteration.

Improvements

The team identified a few potential improvements for further iteration of this product. First, enable the AI to cultivate and maintain a database of memories for preferences and results of previous discussions with the AI and feedback from the user about what they ate and how their dining experience was. This may lead to further refinement of future conversations for more finely tailored recommendations that may not follow the wisdom of crowds.

Next, a centralized repository of dining experiences for everyone in the interested party for the AI to reference. It can take the initial prompt from the primary user and chain pertinent database lookups to meld a dining experience that will fit the most people as appropriate. In order to make decisions about known limitations, the AI might be instructed to use an implied negative voting policy. That is, individual dining preference and limitations will refine the selection of dining choices. In the event of over-constraint of options, the user might be prompted to exclude certain users, or move to a more generalized search of potential restaurants.

Last it may be beneficial to take current location, transportation availability, and time constraints into account when performing a search for restaurants in an area. The current iteration of the program takes into account the locality of the request; however, it cannot pre-determine if the user has the means or ability to get to the chosen location. This may not be an issue for people with personal transportation. If the user is limited to a small location near a convention center, searching for good pizza in New York City may produce a valid response; if it is inaccessible then the response is of no use.

Conclusion

The vacation planning market at the dawn of LLM travel planning is currently a plethora of poorly sourced LLMs that seem to be searching far too wide a field to produce anything useful. Further, once on that vacation, people usually recognize that planned meals might not be right for the party at that time. This project hopes to provide both the most accurate and up-to-date information available and a method for the user to discover the perfect meal for the moment. The solution uses a tech stack consisting of a Python-based chat application that wraps a Gemini LLM with LangChain and two separate APIs, Trip Advisor and Google Maps. After offering

simple instructions and starting a conversation, the AI takes instructions from the user, crafts an API search and looks through both the descriptions of the result and the reviews provided by users and decides if that information fits with the user's prompt. The program then returns several options to the user for them to select from. After the selection, the program then offers the Google Maps location and description of the restaurant to the user, and prompts with the offer of restarting the conversation or refining the search. The final product works well while understanding the hard constraints inherent with crowdsourced information. Further testing will be needed to determine the robustness of the formula with wider and more diverse groups of users and with users that have large parties.

Bibliography

- [1] Anonymous. 2024. Planning a summer vacation? Let AI be your travel guide! *UC Davis IET*. Retrieved from <https://iet.ucdavis.edu/news/planning-summer-vacation-let-ai-be-your-travel-guide>
- [2] J D Biersdorfer. 2025. Can an A.I. Travel Bot Plan Your Trip to NYC? *The New York Times*. Retrieved from <https://www.nytimes.com/2025/03/07/travel/ai-travel-planning-nyc.html>
- [3] Aili Chen, Xuyang Ge, Ziquan Fu, Yanghua Xiao, and Jiangjie Chen. 2024. TravelAgent: An AI Assistant for Personalized Travel Planning. *arXiv.org*. Retrieved from <https://arxiv.org/abs/2409.08069>
- [4] Komal Londhe, Nikita Dharmadhikari, Parth Zaveri, and Unal Sakoglu. 2024. Enhanced Travel Experience using Artificial Intelligence: A Data-driven Approach. *Procedia Computer Science* 235, (January 2024), 1920–1928. DOI:<https://doi.org/10.1016/j.procs.2024.04.182>
- [5] Ankita Mudhale, Madhuri Shirmale, Vedant Kudalkar, Rishikesh Motiray, and Sharique Ahmad. 2024. *Travel Itinerary Planner Using AI*. Retrieved from <https://www.irjet.net/archives/V11/i4/IRJET-V11I4152.pdf>
- [6] Sai Mohith, HemaMalini B H, Karthik K, Nithin Reddy, and Sanath R. 2025. A Review Paper On AI-Driven Travel Planning. *Research Gate*. DOI:<https://doi.org/10.5281/zenodo.14591573>
- [7] Rhiannon Williams. 2024. How to use AI to plan your next vacation. *MIT Technology Review*. Retrieved from <https://www.technologyreview.com/2024/07/08/1094733/how-to-use-ai-to-plan-your-next-vacation/>
- [8] 2024. GuideGeek. *Guidegeek.com*. Retrieved April 4, 2025 from <https://guidegeek.com/>
- [9] 2025. AI Trip Planner | Wonderplan. *Wonderplan*. Retrieved April 4, 2025 from <https://wonderplan.ai/v2/trip-planner>
- [10] 2025. Home. *Usevacay.com*. Retrieved April 4, 2025 from <https://www.usevacay.com/>
- [11] Trip planner AI. *tripplanner.ai*. Retrieved April 4, 2025 from <https://tripplanner.ai/>
- [12] Overview. *Tripadvisor Content API*. Retrieved April 4, 2025 from <https://tripadvisor-content-api.readme.io/reference/overview>
- [13] mindtrip. *mindtrip.ai*. Retrieved April 4, 2025 from <https://mindtrip.ai/>