Individual Project Report: He Mingyang (A0231408N)

Personal Contribution

All of my group mates and I participated in the whole project design and took much effort to address problems we met. I proposed the travelai idea in our confusing time that we do not know which topic to choose, and brainstormed how many functions we should build in this system. In the backend I was responsible for the Hotel recommendation part, more specifically, I tried several APIs and found the most appropriate one, and built some rules to retrieve optimal selections. Besides, I offered a solution to use places of interest API successfully: we first want to retrieve places of interest information based on city name, but we failed to find any API with location name as required input parameter, so i suggested that we can convert location name to its corresponding longitude and latitude first, and finally we accepted this suggestion to finish my places of interest recommendation function. I also made some adjustments to the web page layout so that it looks clearer and more beautiful. Lastly, I was responsible for the System Development & Implementation part of our final project report and all of the Appendix parts.

Learning Journey and Outcome:

I have learnt many things in this practical project, trying to use many tools learnt in Reasoning Systems, Machine Reasoning and Cognitive System before, although some tools are not used in our final system, my group mates and i also make a lot of discussions on whether we should use some tools and how to use them in detail, it is very helpful for us to understand the course deeply. For example, we planed to use Google Dialog Flow as chatbot at first, because we want to build a travelai, so we should consider some ways that can communicate with our users, but with the deepening of our project, we found that Dialog Flow was not flexible enough, more specifically, we need some integer parameters to represent “adults number”, “hotel star rating” etc, but instead of setting a set of rules or setting fixed dialog order, we do not know how to let Dialog Flow distinguish “hotel star rating” and “adults number” in a text, so at last we have to give up Dialog Flow and use NLP model to meet our requirements. Besides, I also have learnt how to use Github to create a repository and write some markdown notes. All these things are new to me, without this project, I will not have the opportunity to use Github before I work officially.

The most important thing is that I learnt how to separate a hard problem apart from many simple problems and solve them. We want to build a travel system that can help people to plan their trips, so we need separate our system into user interface which offer user’s information, database which save all information like flights, hotel, places of interest, a strategy to decide which information need to be retrivalled, and finally wrapped all those functions into an app or a web page. After our effort, we built a system that can give users the optimal flight information, recommend several hotel selections based on the number of adults, star rating etc, and recommend several different kinds of places of interest that give users more ideas to decide where to go in limited travel time.

Knowledge and Skills Application

Cognitive Reasoning System: NLP model
We need a NPL model to extract location information, date information and other integer parameters in different meanings, so we use a pre-trained NLP model and use transfer learning method to finetune the NLP model.

Reasoning System & Machine Reasoning:
We use sorting algorithms and rule-based systems based on users' information and requirements to offer optimal flight information, hotel information and places of interest.

Other Tools: Streamlit, API
API: We use Priceline com Provider API to get location IATA code, flight information. We use Hotel Search API to get hotel information and we use The Travel Advisor API to get places of interest information, all of these 3 APIs are from Rapid API and all of these retrieval are based on "requests.Get" protocols.
Steamlit: we use Streamlit which is a user-friendly way to build web apps. It is all in python, and we use Streamlit to build our own web app which has the function of user interface and displaying our recommendation in this web. We receive users' information from our web page and send it to our backend, and finally return our recommendations to the web page and display them to users.