**Facebook Restaurant Chatbot**

1. Introduction

Whenever you want to find a restaurant, it is often hard to figure out where to start because you sometimes do not know exactly what you want. For example, if you travel to a new city and do not have any background about cuisine, finding a restaurant might be challenging. My Facebook chatbot is created in order to assist users in choosing a suitable restaurant. The main task of the bot is to handle restaurant search when a user asks for suggestions. In response, it will give back a list based on the information the user provides. If the user wants a restaurant nearby, it can locate his/her position and gives results close to the user. Lastly, one small functionality is that the bot can handle greetings, goodbyes or questions about itself and its creator.

2. Related Work

There has been numerous works on implementing a restaurant chatbot. However, most of them are built by specific restaurants to advertise for themselves as well as to give users information only regard to their menu or open hours. For chatbots that assist users in choosing a restaurant, they are usually restricted to **system-initiative**. That is, the bot only gives the user some options to pick and does not understand anything else that they type in. Some of the most famous and related chatbot or way of finding restaurant are:

a) The Food Network Chatbot

One of the most well-known chatbot integrated with Facebook Messenger, *The Food Network Chatbot*, is used to search for food. This chatbot promptly gives users a number of questions to choose such as ingredients or meal type to finally suggest a disk. As mentioned above, this method is simple to build because users and system know exactly what they are about to say next. Despite its success, this approach is too limited.

b) Mica, the Hipster Cat Bot

*Mica, the Hipster Cat Chat Bot* is a venue recommendation bot. In order to utilize Mica, the user sends a location through text e.g. “New York” or "San Francisco" and Mica shows you a selection of the best restaurants or coffee shops close by. If no venue was found, Mica sends you a funny cat picture or asks you a question to entertain you. This app is very similar to my product, but the user does not get to choose their type of food.

c) Search directly in website

The most common way that people nowadays use to find food is to search directly in a website such as Google or Yelp. This is the easiest way and guaranteed to get as many options as possible. However, my chatbot ranks the restaurants by number of reviews for the user. Moreover, it displays necessary information, including price and distance, in a compact way to do a comparison.

3. Methods

First, for any sentence that the user types in, I use regular expression to check for keywords and handle them. For example, in the function isGreeting I check whether users greet by saying e.g “hi” or “hello”. Then I have a bank of response that the bot can randomly picks one to display. The same thing happens when users ask about the bot’s information or say goodbye.

The main task of the bot is to handle finding restaurant. To invoke this functionality, users first give some information about what type of restaurant they want. The information can be anything from simply being like just “restaurant” to detailed request like “sushi bar in Columbus”. When the functionality is invoked, the bot extracts noun phrases and verbs from the user’s command, and then saves them. There are two possibilities when the bot is asked to suggest restaurants. Firstly, if the user specifies their desired location, the bot will use YelpAPI to find restaurants in that location and ranks them based on the number of review. On the other hand, if no location is mentioned, the chatbot assumes that the user wants options close to them. Consequently, it asks the user to send them their location. Using the location, the bot looks for restaurants and ranks them based on how close they are to the user. Similarity, if the user specifies that they want a restaurant nearby, then the same process is executed.

After finish searching, the bot shows the user five best options and associated prices, distance and categories. If the user is attracted to any option, they can simply click on “View details” to look at it more carefully.

4. Evaluation

The chatbot is able to handle command efficiently and accurately. For basic conversation sentences, the process of picking a random response from the bank is very fast. For restaurant search, extracting keywords and calling YelpAPI are also quick and guaranteed to be give best options.

5. Discussion

First of all, the implemented chatbot is good for simplicity, high efficiency and good accuracy. As mentioned above, the process of extracting keywords, picking a response or calling YelpAPI does not take much time to complete. Additionally, Yelp is an extremely famous website with a huge database, therefore the result it provides is trustworthy to the user.

Although this approach can handle requests pretty well, it does not support much interaction with the user. Specifically, the number of shown restaurants is exactly five and the way it ranks the result list is fixed. The user cannot change the number of output as well as how my chatbot ranks them. Additionally, it cannot generate a real conversation with the user. It only focuses on handling one-sentence command. That is, it can only show restaurants based on what the user gives to it. It only asks for location if not specified but cannot promptly ask for more detail like in a real conversation.

For this project, I learn the basic of how bot understands natural language and processes. In the future, I want to make the bot more “real” by allowing the user to manipulate how it outputs the restaurant list. For example, the user should be able to change how many result to be outputted or how the bot ranks them.

6. References