Christos Dimopoulos

C.Dimopoulos@acg.edu

Abstract

Christos Dimopoulos  
ID: 241546  
Instructor: Marios Paraskevopoulos  
Deadline: 21/4/2024

ITC 3161 / voice user interface

Final Project Report

Contents

[Introduction 2](#_Toc164260191)

[Technical Explanation/How it works 3](#_Toc164260192)

[Technical Challenges / Decisions 7](#_Toc164260193)

# Introduction

In the implementation of what we learn in this course this semester, I decided to try creating a chatbot that has to do with food orders. In order to implement this chatbot it was needed first of all to do some research about how all the chatbots of that category are working. And by that, I mean to find first of all how the system will respond on the user when the user asks them of what to order, on how the system will welcome the user when the user comes in my chatbot and wants to take an order or wants to see the menu or just wants to make just an order without any further interaction with the system. After my small research, I found a small guide that I could follow and it is cited [here](https://www.pycodemates.com/2021/11/build-a-AI-chatbot-using-python-and-deep-learning.html). From that guide, I took some ideas on how to make first of all how the system should respond to the variety of questions that user has. Then I created a small json file. This JSON file is named intents, and it has some tags on it. Those tags include greeting, menu, order, take order, thanks, goodbye as long as a tag about how the system will ask the user about their personal info in order to successfully place the order. This JSON also includes all the necessary responses to all of those cases mentioned before.

A screenshot of a computer

Description automatically generated

Here we can have a small idea on how this intents file looks like. Here I want to comment that the menu section has all the menu that my theoretical restaurant has, along with their ingredients as long as the price each one part of the menu has.

After the implementation of the JSON file, I started on writing and modifying the code for the actual chatbot/voice Bot. The above-mentioned article that I took a small idea about the interactions with chat and creating a JSON file includes also a small pretrained model. That model is the base of my whole chatbot project. It includes libraries for understanding the natural – human language and how it works (nltk library) as long as some other libraries that help with machine learning and education and training of neural networks (tensorflow and numpy). After a small thought about what I need and how I want my whole project to work I decided to include the following libraries.

A screen shot of a computer

Description automatically generated

Apart from tensorflow, nltk and numpy, I also use the libraries re, random, the wordnet lemmatizer as long as the speech recognition and pytttsx libraries. The re library is mainly used for regular expressions that used in this program, the random library mainly for the randomness of the responses that the program should respond to the user, the pyttsx3 is used for the conversion of the written words to voice, that means that the text that is written by the program is transformed to voice messages. The speech recognition library is used by the program in order to understand the voice of the user. Finally, the lemmatizer is used from the program to understand what part of speech every word is the user writes or says.

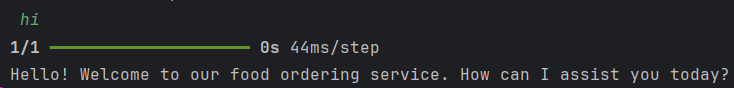
# Technical Explanation/How it works

As I mentioned before, in this project I tried to make a small chatbot regarding food ordering. The initial idea for this project was to only make a program that just expects from the user to say something to it like “hi” or “I want to order” or “gimme the menu” and then the machine to reply to the user something like “hello there” or “what you wanna order”, but this thing would be both simplistic and partly out of context from the scope of the whole course since we learn about how the machine should respond to humans, how all the IVR and similar devices are working. So, I decided to have two choices that the user should choose to interact with the system. The first choice of interaction is just mentioned, it is the chatbot. When the program starts running, the user has the choice to press 1 to select this choice. We can see it below.

A screen shot of a computer

Description automatically generated

In that case the user should start the conversation with our bot either by saying just a hi or asking about the menu.



In the screenshot above we can see how the chat responds to the word ‘hi’. After welcoming the user, the chat immediately asks him how it can help him.

And here is how the system responds to the user on general words prompting him to order something or just to see the menu.

A screenshot of a computer program

Description automatically generated

In this case we can see what our program responds to when our user wants to see the menu.

A screenshot of a computer

Description automatically generated

Here our user ( in that case me) wants 2 pizzas and 3 sandwiches. Then the program responds with what the user’s list includes. In that case the two pizzas and the sandwiches. We can see that the program works both when the user writes something in numbers and numbers written in words. Here also we can see that the program calculates the value of the whole order based on the catalog presented in the actual menu.

A screenshot of a computer

Description automatically generated

Here we can see that the user wants to finish their order. Then the chat prompted him to put their actual info ( telephone, address, and phone number) in order for the order to finish. Then our bot confirms the info of the user as long as what he had on the order.

A black background with white text

Description automatically generated

In the above screenshot we can se the case the user does not want to complete the order so he has to press 0 button to finish.

A black background with white text

Description automatically generated

In a similar manner it works with the second choice, that is the interaction between chat and the new customer. For that choice the user should press 2. In the screenshot below we can see this interaction. As a note here you have to be as clear as possible in order for the program to listen what you actually said. My voice is really low and that’s why the program did not hear me in the first place.

A screenshot of a computer program

Description automatically generated

Here again in the screenshot below we can see the menu again, but this time Google’s voice is reading it. Unfortunately, the new user is obliged to hear the whole menu here along with each value and then place the order. As I will mention in the section of the challenges I met, this voice chat does not have Barge-In in order for the system to stop when the user starts speaking.

A screenshot of a menu

Description automatically generated

The interaction between user and machine is the same as before about the way of ordering food and the successful ordering of food. The only thing that differs is that user has to respond with an exit in order to stop the interaction in the case that he do not want to place the order. Again, you as a user have to be clear. The voice may be misunderstood from the machine as my voice. Here is a screenshot with the exit choice.

A black screen with white text

Description automatically generated

In case he wants to place the order, the screenshot is the following, and then the program terminates. The screenshot:

A screenshot of a computer

Description automatically generated

As a general comment, unfortunately, my code does not accept plurals as its commands, e.g. “I want 2 pizzas” or “I want 3 salads”, so the chatbot/ voice bot will get confused on those commands.

# Technical Challenges / Decisions

The time has come to say what challenges I met when I created this chatbot as long as what decisions I made during this creating trip. This trip was very pleasant for me because I started to explore how a chatbot is made, how I can combine voice along with written text, and also again I had the opportunity to see and understand better how neural networks are made and work. I explored how an API can be interpreted inside a program as long as the program and the API are working together. In that case I used a voice API for the second choice the user has for selecting either chat conversation or voice conversation. The real challenge for me was to put inside my program the barge-in function. This function was really expectable for something that uses voice inside it and interacts with the user. Unfortunately, this task failed successfully to be done because as I was searching how I can use this method to my code I couldn’t find anything that could really help me. I only found little information about that topic on either some sites or Chat – GPT but either what I found was not working at all or every correction I made was useless. Maybe if I had some more time for research, I could put this function working on my program. The general development of this program was simple to make and not so challenging as a whole.