

# Chatbot Project Report - COSC 310 Assignment 3

Nathan Wright

## Project Description:

The project is a modified version of the electron chatbot from Assignment 2. The chatbot's capabilities as a medical advisor/doctor have been slightly improved with the addition of two extra features.

## Github Username and Repository Link:

Nathan Wright - DapperShark1

<https://github.com/DapperShark1/chatbot-app/tree/dev>

(New documentation can be found at the end of the README)

## API's Used:

Google:

- Directions API
- Geocoding API
- Geolocation API
- Places API

Wikipedia API

## Wikipedia Searching - Wikipedia API:

This new feature makes use of the wikipedia API. The user can ask the bot to search for and summarize a wikipedia page using the keywords:

...search for "query"  
...look up "query"  
...wikipedia for "query"

For the bot to work best the query should be the last part of the sentence. This feature allows the user to look up medical information without the need for scrolling through a wikipedia page themselves. There's also a check to see if the query is health related or not. If the bot thinks the query is not health related it will ask for user confirmation. Unfortunately, wikipedia does not tag their meaning; the bot must rely on page categories. Thus this aspect of the wikipedia search feature can be unreliable.

(Image example can be seen in SEARCH.PNG)

## Directions to a location - Google Maps API (geolocation, geocoding, places, and directions):

This new feature makes use of Google's geolocation, and places API. This feature allows the user to ask for a direction to a specific location within 10km of their specified location. The bot

will provide the user with directions to the closest location specified, much like a GPS. This allows the user to ask for directions to their closest clinic, hospital, or other medical facility should they need it. This feature isn't limited to giving directions to just hospitals either.

The user can ask the bot for directions using the keywords:

- ...directions to "location name/description"
- ...directions for "location name/description"
- ...where is "location name/description"
- ...location of "location name/description"
- ...the nearest "location name/description"
- ...the closest "location name/description"

Afterwards the bot will ask the user for their address. Once the address is entered the bot will provide the user directions to their (most likely) desired location. For simplicity's sake the bot assumes the user will be driving.

(Image example can be seen in DIRECT.PNG)

### **The Functions:**

In agent.py two new functions were added:

- wikiSearch(query, base)
- direction(query, base)

Both wikiSearch and direction take in a query and a base parameter. The query parameter is the user's input while the base is the current response the bot believes is appropriate. wikiSearch and direction check to see if any modification is required to the current bots calculated output. If there is editing needed, wikiSearch and direction will modify the bots calculated response. wikiSearch will add any information extracted from wikipedia that the user has requested to search. directions handles calculating the directions the bot should reply with and handling address input as well as confirmation.

### **Limitations:**

- wikiSearch and direction can only be triggered by specific key words
- The functions aren't completely integrated into standard conversation. (The bot won't use a wikipedia article outside of wikiSearch)
- wikiSearch responses may be irrelevant
- direction doesn't preserve the address after entering it for the first time