Joshua Braun

CSCE 240

Project Report

Requirements:

The instructor asked us to create a program titled myrep-chatbot that could interact with users. The chatbot had to handle any input and be able to answer certain questions about a district representative. The questions included personal information, contact information, committee assignments, and what district that was supported. The chatbot also had to be able to provide all information when asked. The instructor also wanted the user to be able to ask about statistics of previous chats such as questions asked and the time it took for each chat. To reach this goal the assignment was broken up into 5 parts. The first assignment was an extractor that collected data from the district of choice. The second assignment was for processing the data and when given an information type program was supposed to return the content. The third assignment was a program that ran in an infinite loop and handled any user response. It had to quit when the user typed “quit”. The program also had to handle known user queries. The fourth assignment was to design a program that could take a user’s utterance and match it to the closest supported query with a confidence estimate. The last assignment was to create a program that would record the chat and report statistics on each session since the chatbot was created. After completing these assignments, we were just supposed to put them together to form the final chatbot.

Scope:

The first decision I made was in selecting the district representative. I chose district five. The language I decided to use was C++. This is because I have already spent considerable time learning Java in other classes and wanted to try and broaden my abilities as a programmer. For extracting the information from the website, I decided to convert the information into a local text file. This is because I could not create a program that would access the website. I then created a function that would access the text file and print all the information to an output file. For processing the information, I took in the output file from the extractor and created two methods. The first method splits a string in two where a semicolon is found. The purpose of this method was so that the user can put in a string containing a broad info type and then a more specific one separated by a semicolon. The program can then separate these so it can more easily find the information. The second method got the info from the output file. It worked by first getting the broad info type requested and searching the output file for this category. Once found it would copy all the information until a blank line was reached because each category was separated by a line. There were also multiple if loops in this method for collecting specific data by searching the broad info type requested using a string stream. For the ui program I created a simple while loop that continually grabbed an input line from the user. If this input matched “quit” the while loop would end. Otherwise, it would call a new method designed to handle any user query. This new method used several different regex patterns to determine what the user was asking. The patterns would match to phrases the computer knew the answer to such as “what is my representative’s contact information” and then call my method for getting the info if it matched. If nothing matched the program would simply say it did not understand and ask the user to rephrase. That way the chatbot could handle any question formatted any way without creating an error. I then broadened this concept in the fourth program by creating a match method that would take two strings as inputs. The two strings would then be compared word by word and a percent match would be calculated. If the percent was over a threshold the program would assume it had the correct question and print an answer. This allowed my program to answer questions even if the user input them incorrectly or had typos. For the final part of the project the program needed to log past sessions and statistics. I did this by instead of just creating an output file, my program would create a file with the current date and time. Then after the user types quit my program would calculate all the statistics of the chat and append them to the end of that file. It would also record those statistics in a csv file. If the user wanted statistics from a session, they could input a command line argument that would access the csv file and print the asked for statistics.

Development Highlights:

The code in my program was done in C++ on the eclipse ide. Eclipse is where the program was tested and ran. To test my code, I input the queries I wanted it to know and determined if the output was correct. I also spelled things wrong and looked to see if the program could still determine the correct answer. Queries with incorrect formatting or random numbers were also input to see if an error was thrown. If I reached an error or the program was not running as I had intended, I would put a cout command into the program to see where in each loop the program was reaching and where it was running into an error. The code was designed in one cpp file without different classes. I did this because I felt that there was no need to create specific classes for this data and instead, I could just use methods for each part of the program.

I faced a variety of problems in creating this program. The first problem I encountered was getting the data from the website. This problem was solved by simply downloading a text version of the website. Another problem was creating the file with the time and then access this file throughout the program to put data into it. I solved this by creating a global variable with the output name and then changing this in a method that created the chat file. Accessing information from the csv file was also difficult and this was solved by using several getline functions with a comma as the delimiter to collect all the information from a line of the csv file. This could then be references when looking for a specific chat’s information.

Reuse:

The code in my program is not very complex. To make it reusable I did not hard code much information. If a user wanted to use my code for a different district’s data, they would simply have to download a text file of the website and then convert it using my extractor. If this is done all my methods should work with that file including my match method that compares two strings. My method that creates an output file using the current time is also very simple and could be reused. My code for the csv file and for searching the chat files is a little less reusable. That code requires specific formatting or else an error could result. All the code used is my own except for certain aspects of some methods where I used online forums to get ideas to perform certain tasks. I would research how to do something in C++ and then modify code someone else had provided to my program’s needs. I do not think anyone has used my code even though my repository is public. The main challenge I faced was getting my program to access outside files and collect data. This was something new for me and it was very difficult to get the right result.

Future Work:

More work can be done to make my chatbot learn over time. A method could be added that makes it so when a user asks a question that my chatbot does not know it could approximate the question and then ask if that is what the user intended. If correct the chatbot could store this information for future reference. More could also be added so that my chatbot seems more personable. Maybe it could answer simple questions about the weather or ask how the user is doing. Another change that could be made is having my program access the information directly from the website. If this was done the input file would not have to be updated periodically and then the program could answer questions about any representative and not just my district. My program could also be changed in the future to collect more data on the representative such as full voting history that way it could answer a variety of more questions. In essence this program is just a starting point from which a more advanced chatbot could be created.