Chatbot Project Report

**System Description**

**Intro and user recognition**

I created a chat bot that talks to users about folding phones. It has a database that stores facts about 6 different folding phones (although only 3 of those have been released so far). The chat bot first asks the user for their name. In this part, named entity recognition is used to determine whether or not the users’ response included a name. If it did, the name they entered is extracted from the text. Otherwise, the bot keeps prompting them for a name until they enter a name. User information is stored in the same database as the facts are. After getting a name from the user, the program checks if that name is already in the database. If it’s not, the user is added to the database. If this is a returning user, the bot will say “Welcome back”, and if any of the user’s preferences are stored in the database, it will make a comment related to one of those.

**Getting the initial topic**

Next, the bot will ask the user which phone they want to talk about. If the user doesn’t know what phones the bot can talk about, and therefore doesn’t pick one that the bot can talk about, the bot will tell the user which phones it has facts about. Another option is the user can say something like “Can you pick one?” and the bot will randomly choose a phone to talk about. For returning users, if the bot knows that they like at least one phone, the bot will choose to talk about one of the phones that they like. This sets the topic for the main loop of the chat bot.

**The main loop**

The first thing that happens in the main loop is the bot reads in the user’s input. First the bot checks to see if the user entered a “special query”, which I will talk about later. If the user did not enter a special query, it then checks to see if it has a reaction to what the user said. To try to make the bot feel more human, if the user says something positive or negative about a phone, the bot will respond with a reaction that could be something like “I like that too!” or “I feel the same way”. There are multiple reactions to positive responses from the user and to negative response from the user, and the bot randomly chooses one. To determine if the user said something positive or negative, sentiment analysis is used. I used the VaderSentiment sentiment analyzer that we talked about in class. If the compound score is above 0.2, the bot will give a positive reaction. If it’s below -0.2, the bot will give a negative reaction. Otherwise, it gives no reaction. I got the number 0.2 from testing VaderSentiment with multiple sentences and determined that it would be a good cutoff. Finally, too make sure that entire conversation isn’t just the bot reacting to what the user says, the bot is not allowed to react 2 times in a row.

If the bot determines it does have a reaction to what the user said or if it just reacted and thus cannot react for this message, it will check if the user wants to talk about a different phone than what the topic is currently set too. If this is the case, the bot will change the topic, and then it will check the rest of the conditions with the new topic. If the user requested the opinion of the bot by saying something like “What’s your opinion?” or “What do you think about this phone?” the bot will give an opinion on these phones. The bot has 3 opinions on each phone which are consistent with each other, and when the user asks for the bot’s opinion, it will randomly choose one it its opinions on the current topic. If the opinion of the bot is not requested, it will then check for specific fact categories. Each fact in the database has a category which can be general, camera, apps, or display. If the user asks for a fact about a phone’s camera, it’s display, or how it handles apps, the bot will retrieve a fact from the corresponding category. Finally, if none of the above conditions are met, the bot will just display a general fact about the current topic.

**User preferences**

The program has a counter for how many facts have been given in a row for a certain phone. Whenever the topic is changed to a different phone, the counter is reset back to 0. When the counter reaches 4, the bot firsts checks to see whether or not it already knows how the user feels about this phone. If it does not, it will ask the user if they like this phone.

VaderSentiment is used to analyze the user’s response. If the user responds with “I don't know”, it will not add anything to the database. If the compound score of the user’s response is above 0, a record is added to the database noting that the user likes that phone. If it is below 0, a record is added to the database noting that the user dislikes that phone. If the user’s response is neutral, nothing is added to the database and the bot informs the user that it will ask again later.

**Special queries**

These are extra things that I added to the chat bot. It would work perfectly fine and meet all the requirements without these, but I felt like they would add more personality to the bot. Since these are extra features, activating them requires the user to exactly match the query, whereas other features such as requesting the bot’s opinion can be activated by many different responses. All of these are case insensitive because the bot converts the user’s responses to lower case. Also, these just have to appear inside the query, they don’t have to be the entire query, so the user may type other things before or after these queries and still activate them.

“goodbye” – A more graceful way to end the conversation then by pressing ^C. This will prompt the bot to say goodbye to you, and then the program terminates.

“what phones do I like” – This will make the bot list out all of the phones that it believes the user likes. It does this by retrieving that users records from the database.

“what phones do I dislike” or “what phones do I not like” – Same as above, except it lists the phones that the user does not like.

“what is your favorite folding phone” or “what is your favorite phone” – Just in case your curious about the bot’s opinions, it will tell you what its favorite folding phone is.

“what is your least favorite folding phone” or “what is your least favorite phone” – The bot will tell you which of these phones it likes the least. In order to be fair, its only considering phones that have been released.

“what is your name” – If you forget the bot’s name, don’t worry! You won’t make it feel bad, just ask and it will remind you.

“you just said that” or “you already said that” – Since the bot randomly chooses facts, sometimes it will choose the same fact two times in a row. When that happens, you can say one of these two things, and the bot will apologize for being repetitive.

**NLP techniques**

Named entity recognition – This is used to identify the name of the user. This way, when the bot asks the user for their name, the user can respond with “John” or “Hello, nice to meet you. My name is John”, and the program can determine that “John” is the users name. I used spaCy’s NER feature. When asking for the user’s name, I used spaCy’s NLP function on the user’s response, and then looked for a token with the PERSON label. This label is what spaCy uses for names. If no token with the PERSON label was found, the bot prompts the user for their name again.

Sentiment analysis – As explained above, I used VaderSentiment for two parts of this project. The first bot was for bot reactions. If the user says something positive enough or negative enough about a phone, the bot will have a reaction to it. I used the compound score to get a good measure of the overall sentiment of the user’s response. For responses with a compound score above 0.2 the bot had a positive reaction. For responses with a score below -0.2 the bot had a negative reaction. The other place where I used sentiment analysis was for determining whether or not the user liked a phone. After the bot asks a user if they like a phone, sentiment analysis is performed on their response. If the compound score is positive, we say the user likes the phone, and the database is updated to reflect that. If the compound score is negative, we say the user dislikes the phone, and the database is updated to reflect that.

Information extraction – Term frequency was used to determine the structure of the database. The database was built up from articles retrieved from the internet. I used a web crawler to find these articles and retrieve the text from them. After cleaning up the text, term frequency was used to determine the top 40 words from these documents. Looking through those words I noticed two main types of words that were actually related to the topic, names of phones or companies that make phones, and aspects of theses phones that these articles talked about a lot. I decided to make a table for each phone that made this top 40 list, and categorize the facts based on the terms that I believed were the most important: camera, display/screen, and apps.

**Important notes for running this chatbot**

This chatbot was created from scratch, so all of the code is contained inside a python program. However, I make use of libraries that you may or may not have downloaded for this class, so to ensure that this program runs, please do the following:

Install spaCy:

This can be done by simply running the command

pip3 install spacy

Download the model:

You will also need to download the spaCy model that I used by running the command

python3 -m spacy download en\_core\_web\_sm

Install VaderSentiment:

I’m pretty sure you had to do this for one of the quizzes in this class, but just in case

pip3 install VaderSentiment

Finally, make sure that the database file (facts.db) is in the same directory as the python file.

**Dialogue Tree**

**Diagram

Description automatically generated**

**Sample Dialogue**

Foldy> Hello, I'm Foldy, the folding phone chat bot! What's your name?

> Jason

Foldy> Welcome back Jason! Would you like to talk about the Galaxy Fold some more?

> I would love too

Foldy> Multitasking poses no problem, and games like Pako Forever, Alto’s Odyssey, and PUBG: Mobile run well on the 7.3-inch screen.

> How is the camera?

Foldy> It’s not the best camera system, but the six cameras on the Galaxy Fold will satisfy, and its versatility means you have more ways than one to take a photo.

> Ok. That's good to know

Foldy> Yeah, it's really awesome!

> What do you think about this phone?

Foldy> If robots could have smartphones, I would buy this one!

> Let's talk about the Mate X

Foldy> On one side of the mate x is a bar which contains components, the camera lenses, and the power button which also contains a fingerprint sensor.

> What else can you tell me about it?

Foldy> The falcon hinge used in the mate x is a mechanical one, and you’ll need to push a button on the bar running down the side to open the device up.

> How does it handle apps?

Foldy> Video playing on the large unfolded screen quickly shifted to the main screen when it was folded up, and then to the rear screen when the Mate X was turned over.

> Sounds like its pretty smart

Foldy> I love it!

> What phones do I like?

Foldy> If I recall correctly, these are the phones that you like:

Samsung Galaxy Fold

TCL folding phones

> Thank you! Goodbye

Foldy> Goodbye! It was nice talking to you.

**Knowledge Base**

The knowledge base contains a table for each of the phones it holds facts about. It also has two tables for the user models.

Graphical user interface, text

Description automatically generated

Each table consists of multiple facts. Each fact is given a category, which is either camera, display, apps, or general.

Graphical user interface, text, application, Word

Description automatically generated

Less is known about the phones that have not been released yet, so those tables have less facts.

Graphical user interface, text, application, Word

Description automatically generated

**User Models**

Two of the tables in the database are used to store user data. The users table is just a list of the names of users. This table is used to check whether or not a user has talked to the chatbot before.

A screenshot of a computer screen

Description automatically generated

The user\_preferences table stores the preferences of the users. It has three attributes: name, which is the user’s name, phone, which is the phone the preference is about, and preference, which is either like or dislike.

A screenshot of a computer screen

Description automatically generated

**Evaluation**

Overall, I would say that this chatbot does a pretty decent job of having a seemingly human conversation, but it’s not perfect. Since the facts were all taken from online articles, some of them don’t really fit the flow of the conversation very well. Also, with the bot choosing random reaction to what the user says, sometimes the reactions don’t fit very well with what the user said. Also, the sentiment analysis is not perfect. When the bot asks the user if they like a phone, if the user responds with “I don’t think it’s very good”, VaderSentiment assigns that sentence a negative score of 0, and a compound score of 0.4927, so the bot will think the user likes this phone, when they clearly don’t. Despite these short comings, I still think the bot is effective at its job. It is able to inform people about folding phones, and for the most part does a good job at determining what the user wants. It knows when the user wants to change the topic. It’s able to recognize when user’s want more specific facts like facts about the camera. It even has features like reactions and giving its own opinion to make it seem more natural. And when asked if they like a phone, I believe most people would answer no by saying “no” or “I don’t like it”, both of which are properly assigned a negative score. So, for most cases, the bot does what it is supposed to do, and it does it in a way that doesn’t feel too unnatural.