Hamna Mustafa

Hbm170002

CS4395 – Human Language Technologies

November 13th 2022

Chatbot Report

I created a chatbot using the information retrieval method. At a very high level, I created a knowledge base and used NLP techniques to format and retrieve queries. I then connected my chatbot to a channel in an IRC (Internet Relay Chat) server known as Libera Chat where users could log on and converse with it.

Running Instructions:

First run KnowledgeBaseTrain.py using the following command:

```
python3 KnowledgeBaseTrain.py
```

Then run main.py:

```
python3 main.py
```

This will connect the chatbot to the IRC server. You can then go to this link:

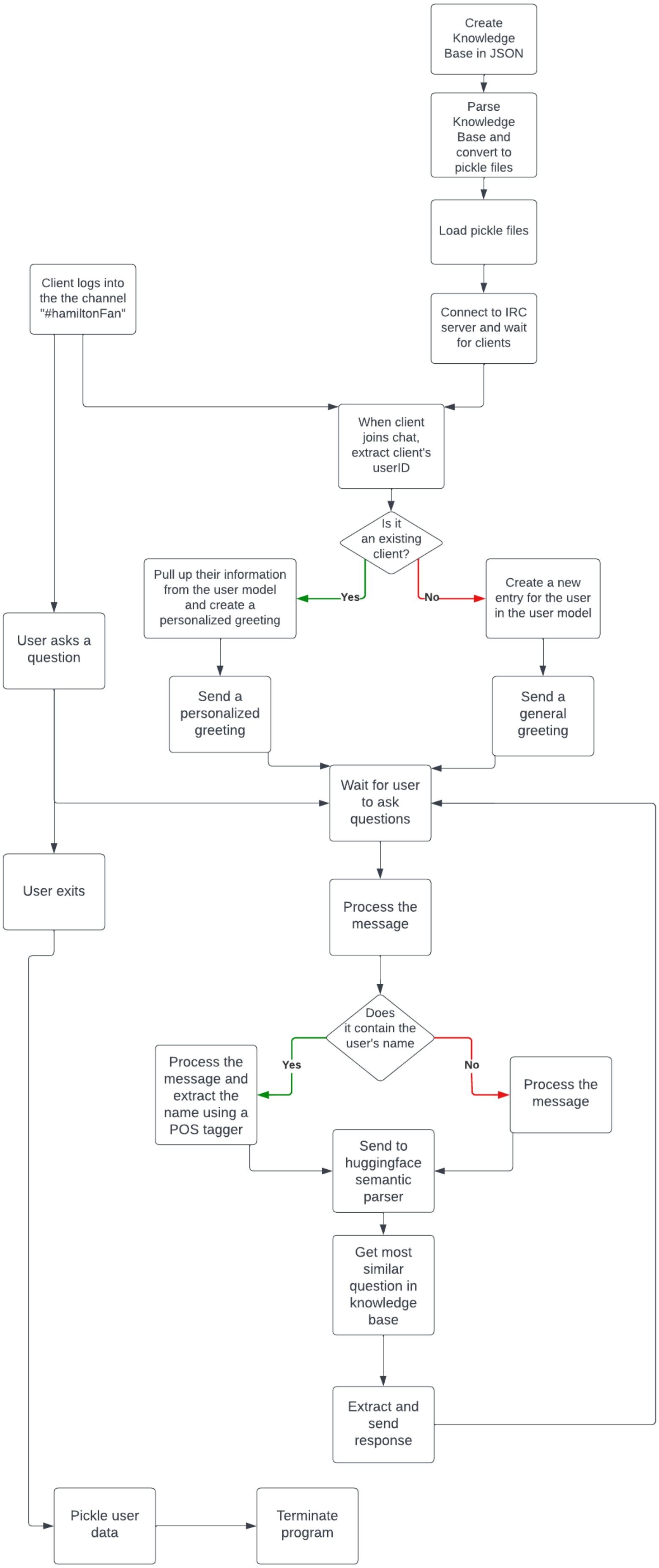
https://web.libera.chat

Then join the channel “#hamiltonFan

Now you can start chatting! Enjoy!

System Description

This project had a number of modules including the knowledge base, information retrieval, processing queries, user models, and an irc server. Below is the logic tree of the program:


Create Knowledge Base in JSON
Parse Knowledge Base and convert to pickle files
Load pickle files
Client logs into the the channel "#hamiltonFan"
Connect to IRC server and wait for clients
When client joins chat, extract client's userID
Is it an existing client?
Yes
No
Pull up their information from the user model and create a personalized greeting
Create a new entry for the user in the user model
User asks a question
Send a personalized greeting
Send a general greeting
Wait for user to ask questions
User exits
Process the message
Does it contain the user's name
Yes
No
Process the message and extract the name using a POS tagger
Process the message
Send to huggingface semantic parser
Get most similar question in knowledge base
Extract and send response
Pickle user data
Terminate program

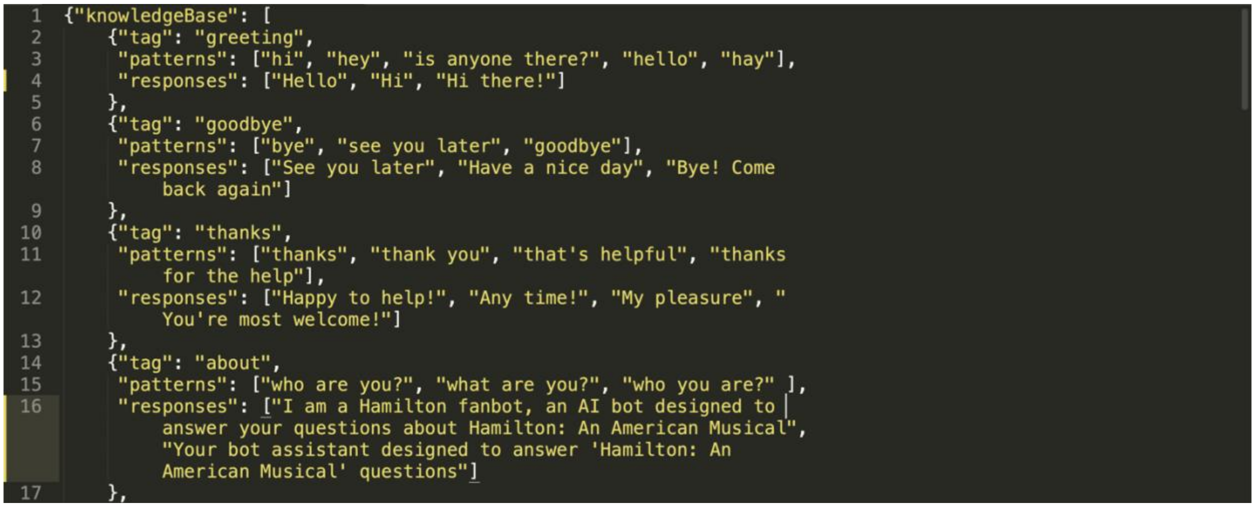
Knowledge Base:

Since my chatbot used the information retrieval method, it needed a knowledge base to retrieve information from. I decided to use the same topic as my web crawler project for this chatbot which was about “Hamilton: An American Musical”. Hence, my chatbot would be a fanbot for Hamilton, providing the user with information regarding the musical.

In order to make it easy for my chatbot to retrieve information, I decided to create a new knowledge base that was in JSON format. I divided the information up by tags, created some example questions, and wrote out several plausible responses to those questions.

Thus, each entry in the knowledge base had a tag, patterns (example queries), and responses.

Here is a screenshot of what the knowledge base looks like:



```
{"knowledgeBase": [
    {"tag": "greeting",
     "patterns": ["hi", "hey", "is anyone there?", "hello", "hay"],
     "responses": ["Hello", "Hi", "Hi there!"]
    },
    {"tag": "goodbye",
     "patterns": ["bye", "see you later", "goodbye"],
     "responses": ["See you later", "Have a nice day", "Bye! Come
         back again"]
    },
    {"tag": "thanks",
     "patterns": ["thanks", "thank you", "that's helpful", "thanks
         for the help"],
     "responses": ["Happy to help!", "Any time!", "My pleasure", "
         You're most welcome!"]
    },
    {"tag": "about",
     "patterns": ["who are you?", "what are you?", "who you are?" ],
     "responses": ["I am a Hamilton fanbot, an AI bot designed to
         answer your questions about Hamilton: An American Musical",
         "Your bot assistant designed to answer 'Hamilton: An
         American Musical' questions"]
    },
```

Once this JSON file was created, I parsed the file and extracted all the patterns along with the corresponding tags and put them into 2 arrays. So, one array had all the queries and the array contained the tags corresponding to those queries. For example:

training_sentences = [“hi”, “hey”, “is anyone there”, “who are you”, “see you later”]
training_label = [“greeting”, ”greeting”, “greeting”, “about”, “goodbye”]

Thus, you can see the the first three phrases in ‘training_sentence’ had the tag ‘greeting’ and so on. I then pickled these arrays into 2 separate pickle files. These files became the part of the knowledge base that my chatbot would directly interact with.

Retrieving Information:

Now that I had a knowledge base to work from, I needed to be able to take the messages sent by the user and retrieve the most relevant information from the knowledge base. It took me a while to figure out how to approach this. I first tried a machine learning approach where the tags were the target and the traning_sentences were the predictors. I tried many different models such as Naïve Bayes, Logistic Regression, and Neural networks but no matter how many models and topologies I tried, I was getting terrible accuracies. Eventually, I realized that there were two major reasons for this: 1) I didn’t have enough data in my knowledge base 2) The queries needed some sort of semantic parsing. Once I realized the second reason, I knew I had to find some way to semantically compare the user’s message with all the data in the knowledge base to find the question most similar in meaning and retrieve the answer. This led me to Huggingface’s “Sentence Similarity”. They had an API that would take the user message and the array of training sentences and return an array which contained the semantic similarity values between the message and the training sentences. For example, if the user message was: “*That is a happy person*” and the training sentences were: [“*That is a happy dog*, “*That is a very happy person*”, “*Today is a sunny day*”], then the Sentence Similarity API would return: [0.853, 0.981, 0.655]. As you can see, “*That is a very happy person*” got the highest score which makes sense because it is the most semantically similar to the user message.

Using this method proved to be very fruitful. Once I would receive all the semantic similarity values, I would find the max value and the corresponding training sentence. I would then pick a response from the knowledge base of that training sentence.

Processing Queries:

Once I had figured out how to find relevant answers, I had to process the user's message to increase the chances of correct retrievals. I used nltk to do this. The processing that I was most concerned with was when the user message contained the user's name. I will discuss the user model later but all you need to know for now is that I needed to be able to extract the user's name. If the user sent this message: "*My name is Hamna*", I needed to be able to parse it and find the name. I used two things to do this: nltk's POS tagger and a variable prevTag. PrevTag was a variable that contained the tag of the previous user message. The way that my knowledge base is designed, if the user asks the bot it's name, the bot responds and asks the user for their name.

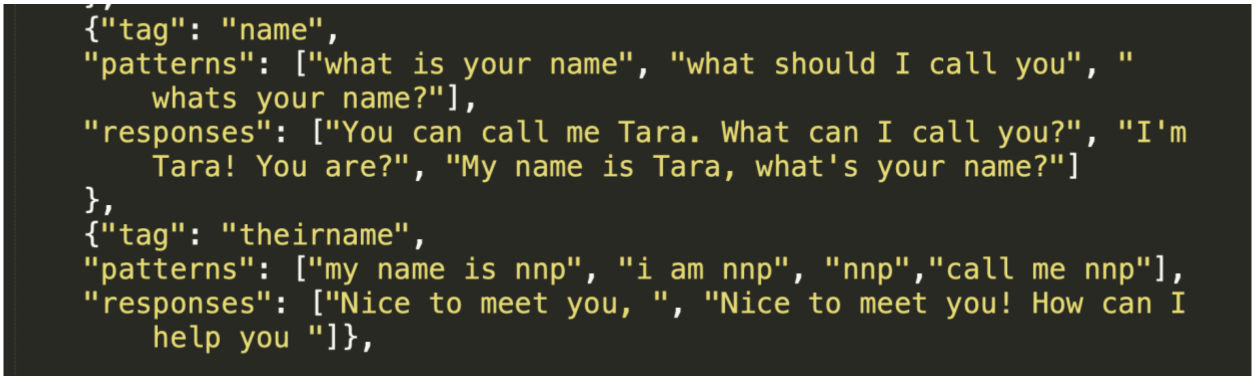
For example:

User: "*What should I call you?*"

Bot: "*You can call me Tara. What can I call you?"*

User: "*My name is Hamna*"

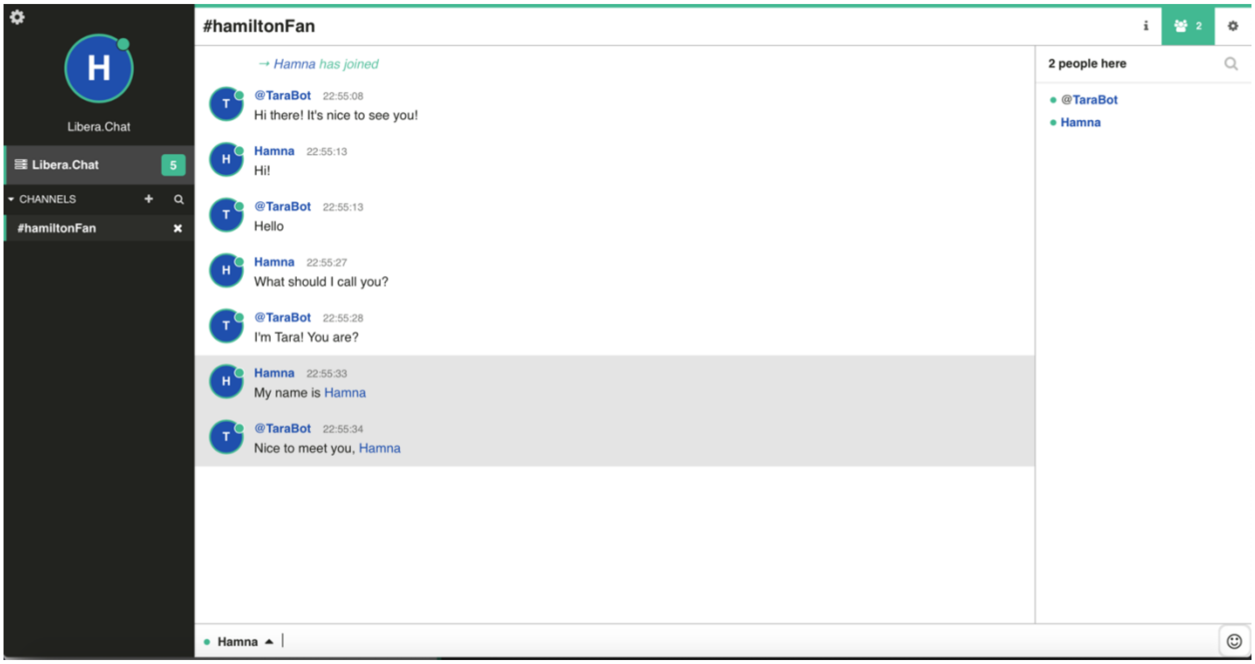
Thus, the first user message fell into the tag "name" and I needed to make sure the second one fell into the tag "theirname". I couldn't directly send the message to the Semantic Similarity parser because every user's name would be different and that would increase the chance for error. Thus, in the training sentences for "theirname", I replaced where the name would be by "nnp". Then, if the prevTag was "name", I used nltk's POS tagging to find the proper noun (NNP) in the user's sentence and replace it with 'nnp'. So if the message was: "*My name is Hamna*", it would be formatted to become "*My name is nnp*". I also saved the name so that it could be used for the user model (which I will talk about in detail later) and so that I could add it to the response string.



```
    {"tag": "name",
    "patterns": ["what is your name", "what should I call you", "
        whats your name?"],
    "responses": ["You can call me Tara. What can I call you?", "I'm
        Tara! You are?", "My name is Tara, what's your name?"]
    },
    {"tag": "theirname",
    "patterns": ["my name is nnp", "i am nnp", "nnp","call me nnp"],
    "responses": ["Nice to meet you, ", "Nice to meet you! How can I
        help you "]},
```

As you can see in the "theirname" responses, it leaves space for the user's name to be concatenated. So if the first response was chosen for the conversation above, the message sent to the user would be: "*Nice to meet you, Hamna*".

Finally, I converted all the user messages to lowercase (except for names) before sending them to huggingface.



#hamiltonFan

→ Hamna has joined

@TaraBot 22:55:08
Hi there! It's nice to see you!

Hamna 22:55:13
Hi!

@TaraBot 22:55:13
Hello

Hamna 22:55:27
What should I call you?

@TaraBot 22:55:28
I'm Tara! You are?

Hamna 22:55:33
My name is Hamna

@TaraBot 22:55:34
Nice to meet you, Hamna

User Model:

As per the project’s description, I had to save each user’s information which could later be used to personalize the chatbot’s responses according to each user. I decided to use a pickled dictionary to do this, with the key being the user’s id. When the program starts, it unpickles the “userData.pickle” if it exists which contains information about users from previous runs. This is the format of each user’s entry:

{

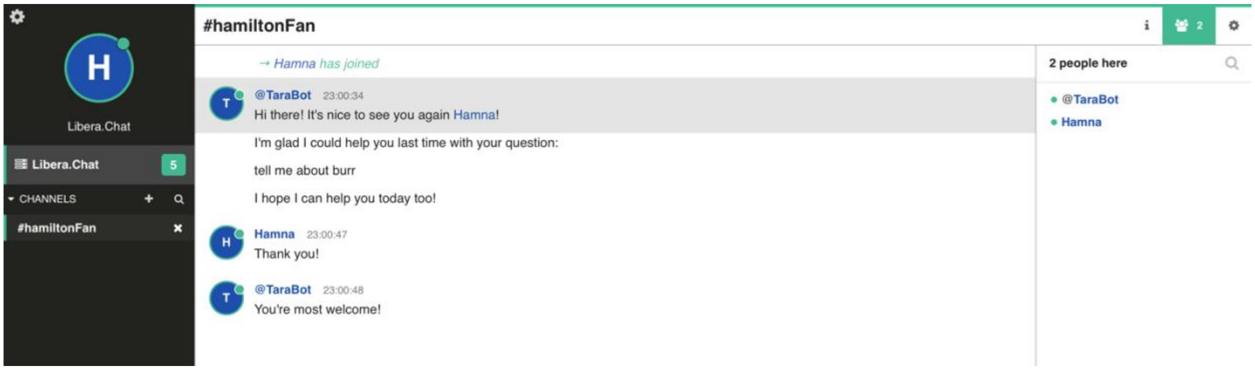
“name”: “”

“queries”: []

“responses”: []

}

Everytime a user joins the chat, the program isolates the user’s ip address and saves it as userid. It then checks if that userid is already in the user model. If it is, then the bot retrieves that information and sends the user a personalized greeting. It uses the user’s name in the greeting and also mentions the last question the user asked.



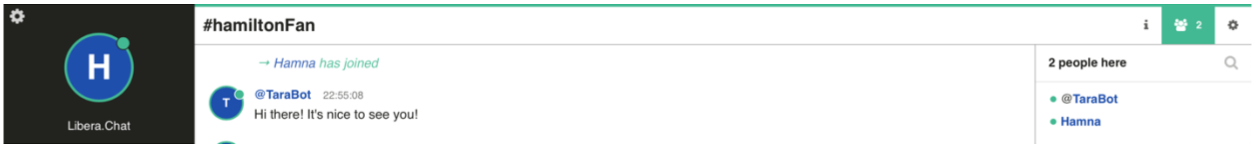
→ Hamna has joined

@TaraBot 23:00:34
Hi there! It's nice to see you again Hamna!
I'm glad I could help you last time with your question:
tell me about burr
I hope I can help you today too!

Hamna 23:00:47
Thank you!

@TaraBot 23:00:48
You're most welcome!

If it is a new user that does not yet exist in the user model, the bot creates an entry for that user which is then filled as the user asks questions and provides information. For the queries and responses, I only stored queries relative to Hamilton and not general ones such as “*Hello*” or “*What’s your name*”. For new users, the bot sends a general greeting.



→ Hamna has joined

@TaraBot 22:55:08
Hi there! It's nice to see you!

- @TaraBot
- Hamna

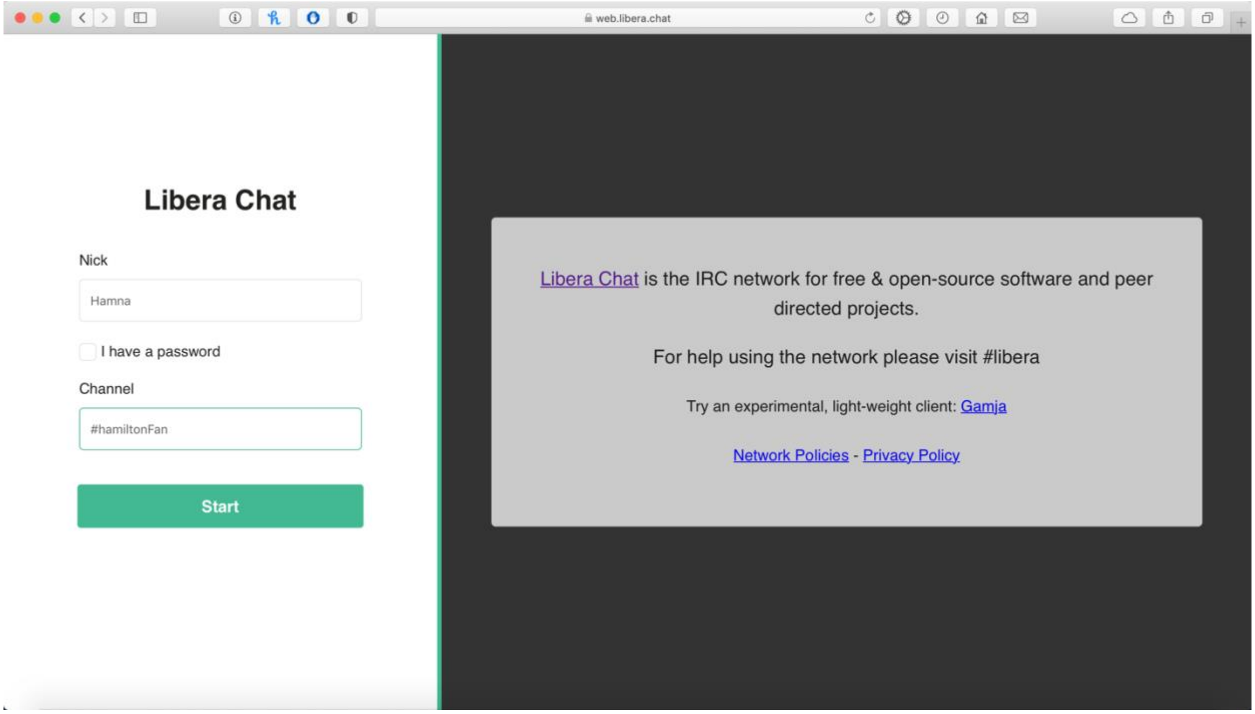
Once the user exits the channel, the bot once again pickles the user model to be used for later runs. It then terminates well.

IRC Server:

Simply writing the bot on my local machine and using the console to converse with it was not enough, I needed to be able to host the bot somewhere so that it could communicate with different users. I decided to connect it to an IRC server using the socket library. The IRC server that I chose was Libera Chat which can be accessed by the following url:

https://web.libera.chat

The channel that I connected to was “#hamiltonFan”. Thus, in order to chat with my bot, all the user needs to do is go to the above url and connect to the channel “#hamiltonFan”.



Libera Chat

Nick

Hamna

☐ I have a password

Channel

#hamiltonFan

Libera Chat is the IRC network for free & open-source software and peer directed projects.

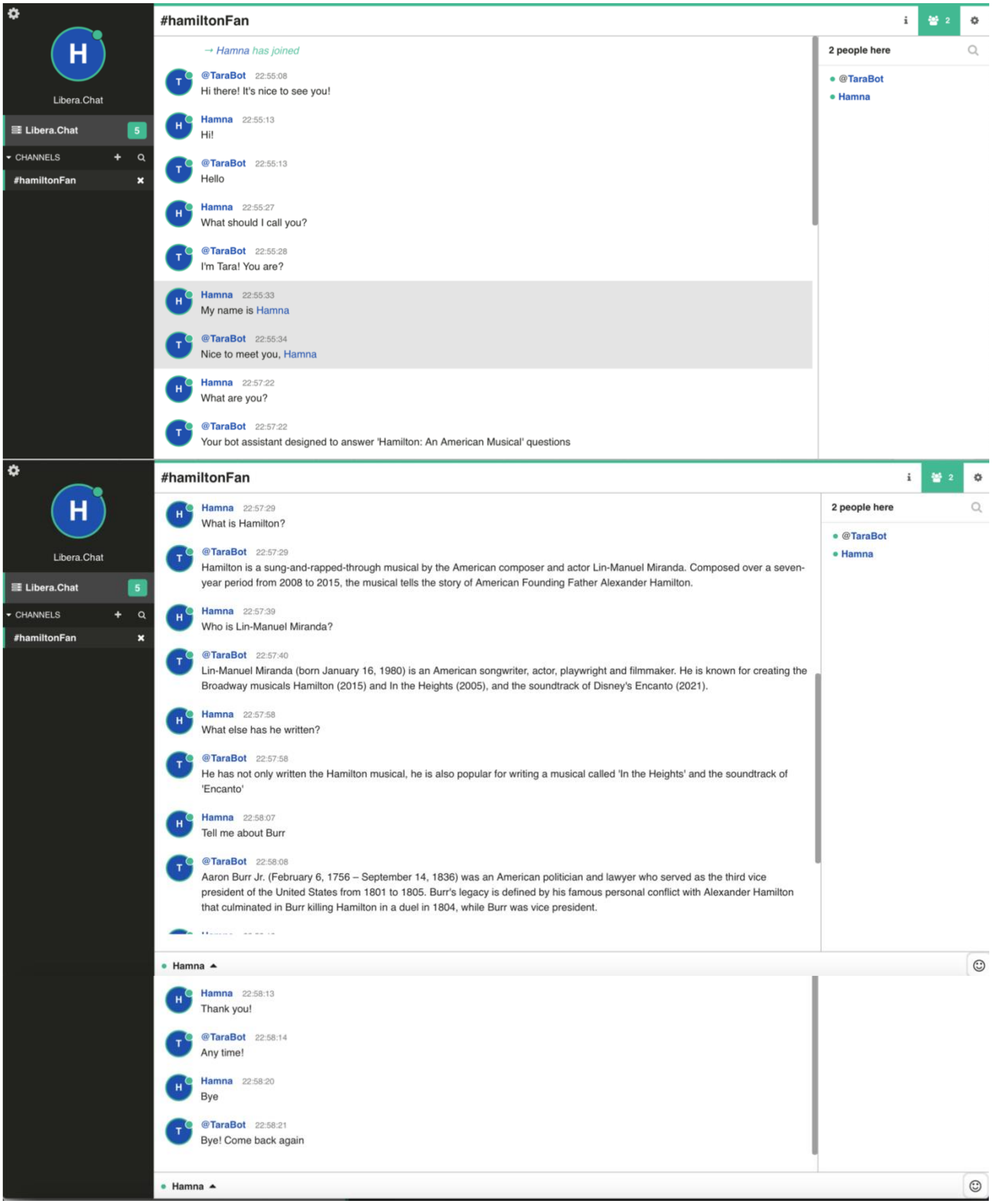
For help using the network please visit #libera

Try an experimental, light-weight client: Gamja

Network Policies - Privacy Policy

The bot connects to this channel when I run the program and stays connected until a user leaves after which it terminates.

Sample Dialog Interaction



→ *Hamna has joined*

@TaraBot 22:55:08
Hi there! It's nice to see you!

Hamna 22:55:13
Hi!

@TaraBot 22:55:13
Hello

Hamna 22:55:27
What should I call you?

@TaraBot 22:55:28
I'm Tara! You are?

Hamna 22:55:33
My name is Hamna

@TaraBot 22:55:34
Nice to meet you, Hamna

Hamna 22:57:22
What are you?

@TaraBot 22:57:22
Your bot assistant designed to answer 'Hamilton: An American Musical' questions

Hamna 22:57:29
What is Hamilton?

@TaraBot 22:57:29
Hamilton is a sung-and-rapped-through musical by the American composer and actor Lin-Manuel Miranda. Composed over a seven-year period from 2008 to 2015, the musical tells the story of American Founding Father Alexander Hamilton.

Hamna 22:57:39
Who is Lin-Manuel Miranda?

@TaraBot 22:57:40
Lin-Manuel Miranda (born January 16, 1980) is an American songwriter, actor, playwright and filmmaker. He is known for creating the Broadway musicals Hamilton (2015) and In the Heights (2005), and the soundtrack of Disney's Encanto (2021).

Hamna 22:57:58
What else has he written?

@TaraBot 22:57:58
He has not only written the Hamilton musical, he is also popular for writing a musical called 'In the Heights' and the soundtrack of 'Encanto'

Hamna 22:58:07
Tell me about Burr

@TaraBot 22:58:08
Aaron Burr Jr. (February 6, 1756 – September 14, 1836) was an American politician and lawyer who served as the third vice president of the United States from 1801 to 1805. Burr's legacy is defined by his famous personal conflict with Alexander Hamilton that culminated in Burr killing Hamilton in a duel in 1804, while Burr was vice president.

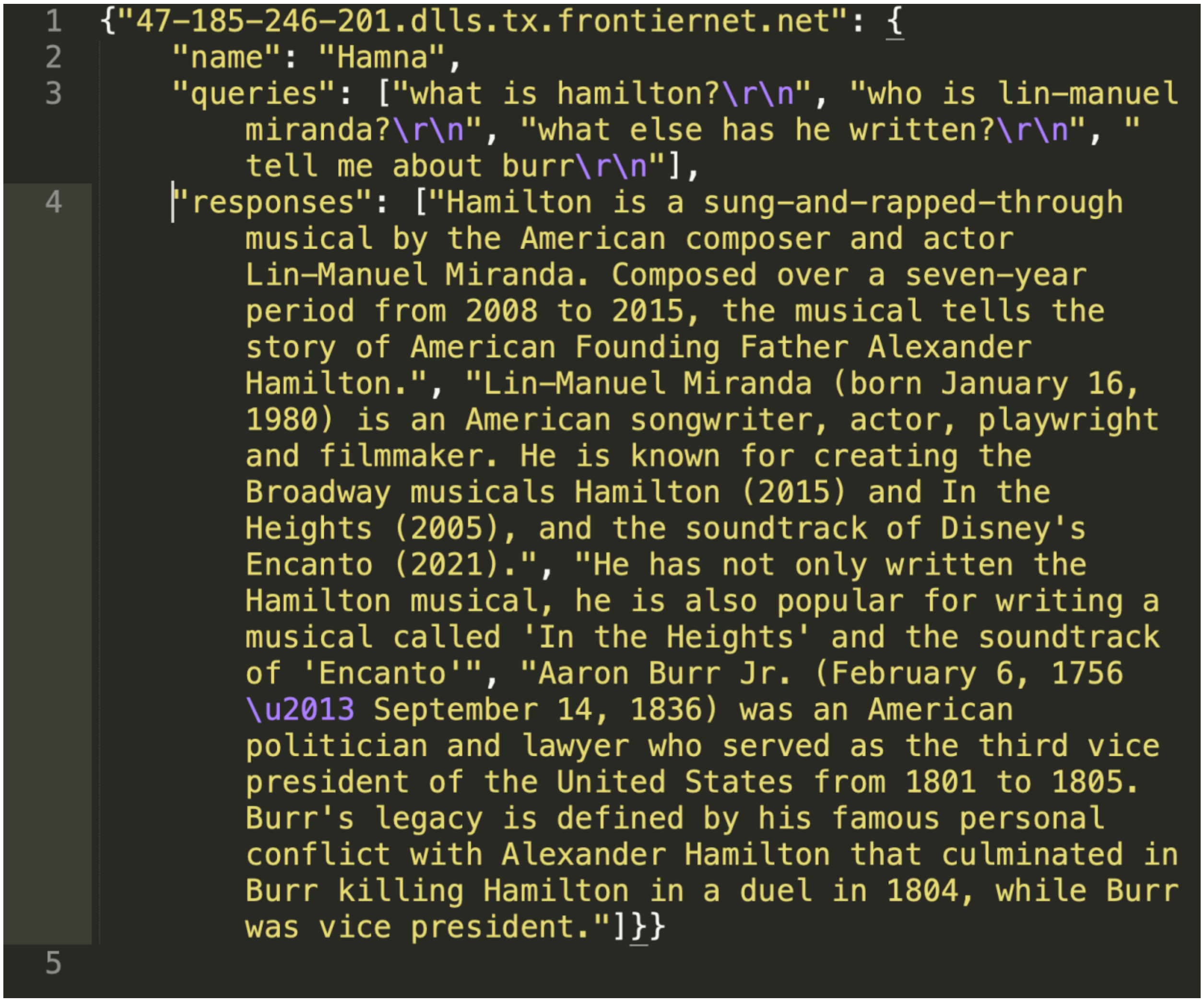
Hamna 22:58:13
Thank you!

@TaraBot 22:58:14
Any time!

Hamna 22:58:20
Bye

@TaraBot 22:58:21
Bye! Come back again

User Model after the above interaction



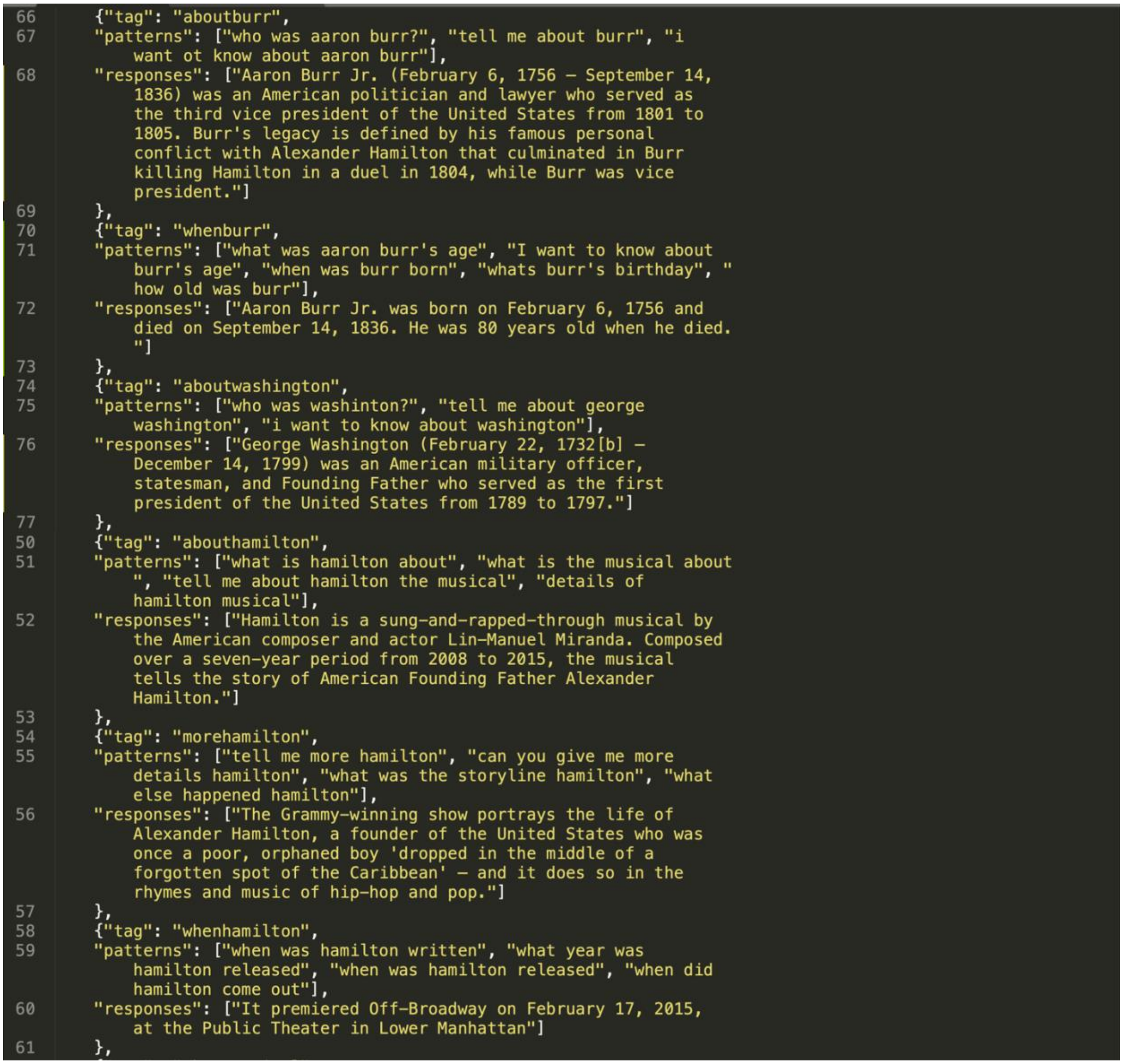
```
{"47-185-246-201.dlls.tx.frontiernet.net": {
    "name": "Hamna",
    "queries": ["what is hamilton?\r\n", "who is lin-manuel
        miranda?\r\n", "what else has he written?\r\n", "
        tell me about burr\r\n"],
    "responses": ["Hamilton is a sung-and-rapped-through
        musical by the American composer and actor
        Lin-Manuel Miranda. Composed over a seven-year
        period from 2008 to 2015, the musical tells the
        story of American Founding Father Alexander
        Hamilton.", "Lin-Manuel Miranda (born January 16,
        1980) is an American songwriter, actor, playwright
        and filmmaker. He is known for creating the
        Broadway musicals Hamilton (2015) and In the
        Heights (2005), and the soundtrack of Disney's
        Encanto (2021).", "He has not only written the
        Hamilton musical, he is also popular for writing a
        musical called 'In the Heights' and the soundtrack
        of 'Encanto'", "Aaron Burr Jr. (February 6, 1756
        \u2013 September 14, 1836) was an American
        politician and lawyer who served as the third vice
        president of the United States from 1801 to 1805.
        Burr's legacy is defined by his famous personal
        conflict with Alexander Hamilton that culminated in
        Burr killing Hamilton in a duel in 1804, while Burr
        was vice president."]}}
```

The user model is actually stored as a pickle file. I converted it to JSON in the above screenshot for easy in reading as pickle files are in binary.

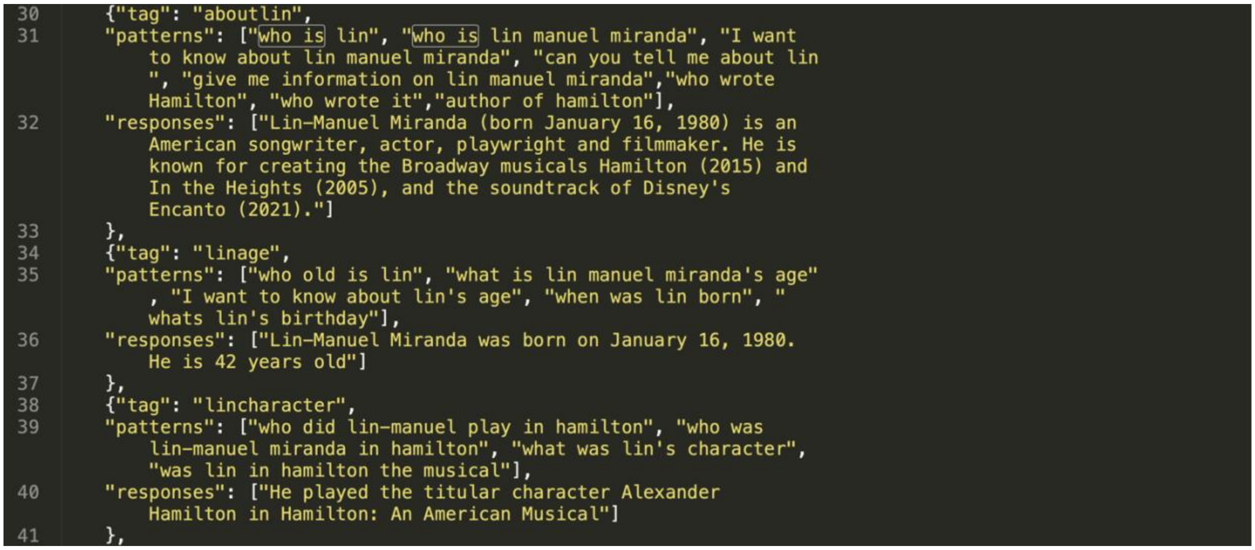
Appendix

Knowledge Base

Although I have attached examples of what the knowledge base looks like above, here are a few more.



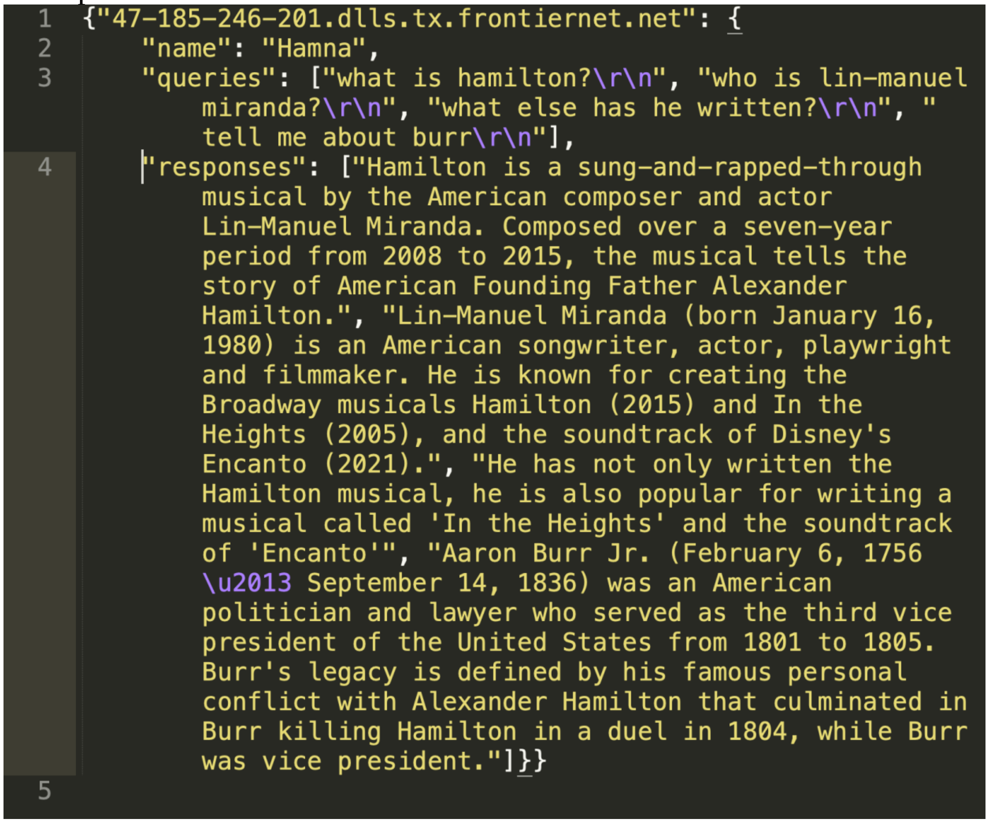
```
    {"tag": "aboutburr",
    "patterns": ["who was aaron burr?", "tell me about burr", "i
        want ot know about aaron burr"],
    "responses": ["Aaron Burr Jr. (February 6, 1756 – September 14,
        1836) was an American politician and lawyer who served as
        the third vice president of the United States from 1801 to
        1805. Burr's legacy is defined by his famous personal
        conflict with Alexander Hamilton that culminated in Burr
        killing Hamilton in a duel in 1804, while Burr was vice
        president."]
    },
    {"tag": "whenburr",
    "patterns": ["what was aaron burr's age", "I want to know about
        burr's age", "when was burr born", "whats burr's birthday", "
        how old was burr"],
    "responses": ["Aaron Burr Jr. was born on February 6, 1756 and
        died on September 14, 1836. He was 80 years old when he died.
        "]
    },
    {"tag": "aboutwashington",
    "patterns": ["who was washinton?", "tell me about george
        washington", "i want to know about washington"],
    "responses": ["George Washington (February 22, 1732[b] –
        December 14, 1799) was an American military officer,
        statesman, and Founding Father who served as the first
        president of the United States from 1789 to 1797."]
    },
    {"tag": "abouthamilton",
    "patterns": ["what is hamilton about", "what is the musical about
        ", "tell me about hamilton the musical", "details of
        hamilton musical"],
    "responses": ["Hamilton is a sung-and-rapped-through musical by
        the American composer and actor Lin-Manuel Miranda. Composed
        over a seven-year period from 2008 to 2015, the musical
        tells the story of American Founding Father Alexander
        Hamilton."]
    },
    {"tag": "morehamilton",
    "patterns": ["tell me more hamilton", "can you give me more
        details hamilton", "what was the storyline hamilton", "what
        else happened hamilton"],
    "responses": ["The Grammy-winning show portrays the life of
        Alexander Hamilton, a founder of the United States who was
        once a poor, orphaned boy 'dropped in the middle of a
        forgotten spot of the Caribbean' – and it does so in the
        rhymes and music of hip-hop and pop."]
    },
    {"tag": "whenhamilton",
    "patterns": ["when was hamilton written", "what year was
        hamilton released", "when was hamilton released", "when did
        hamilton come out"],
    "responses": ["It premiered Off-Broadway on February 17, 2015,
        at the Public Theater in Lower Manhattan"]
    },
```



```
    {"tag": "aboutlin",
    "patterns": ["who is lin", "who is lin manuel miranda", "I want
        to know about lin manuel miranda", "can you tell me about lin
        ", "give me information on lin manuel miranda","who wrote
        Hamilton", "who wrote it","author of hamilton"],
    "responses": ["Lin-Manuel Miranda (born January 16, 1980) is an
        American songwriter, actor, playwright and filmmaker. He is
        known for creating the Broadway musicals Hamilton (2015) and
        In the Heights (2005), and the soundtrack of Disney's
        Encanto (2021)."]
    },
    {"tag": "linage",
    "patterns": ["who old is lin", "what is lin manuel miranda's age"
        , "I want to know about lin's age", "when was lin born", "
        whats lin's birthday"],
    "responses": ["Lin-Manuel Miranda was born on January 16, 1980.
        He is 42 years old"]
    },
    {"tag": "lincharacter",
    "patterns": ["who did lin-manuel play in hamilton", "who was
        lin-manuel miranda in hamilton", "what was lin's character",
        "was lin in hamilton the musical"],
    "responses": ["He played the titular character Alexander
        Hamilton in Hamilton: An American Musical"]
    },
```

User Model

Example of User Model



```
{"47-185-246-201.dlls.tx.frontiernet.net": {
    "name": "Hamna",
    "queries": ["what is hamilton?\r\n", "who is lin-manuel
        miranda?\r\n", "what else has he written?\r\n", "
        tell me about burr\r\n"],
    "responses": ["Hamilton is a sung-and-rapped-through
        musical by the American composer and actor
        Lin-Manuel Miranda. Composed over a seven-year
        period from 2008 to 2015, the musical tells the
        story of American Founding Father Alexander
        Hamilton.", "Lin-Manuel Miranda (born January 16,
        1980) is an American songwriter, actor, playwright
        and filmmaker. He is known for creating the
        Broadway musicals Hamilton (2015) and In the
        Heights (2005), and the soundtrack of Disney's
        Encanto (2021).", "He has not only written the
        Hamilton musical, he is also popular for writing a
        musical called 'In the Heights' and the soundtrack
        of 'Encanto'", "Aaron Burr Jr. (February 6, 1756
        \u2013 September 14, 1836) was an American
        politician and lawyer who served as the third vice
        president of the United States from 1801 to 1805.
        Burr's legacy is defined by his famous personal
        conflict with Alexander Hamilton that culminated in
        Burr killing Hamilton in a duel in 1804, while Burr
        was vice president."]}}
```

Evaluation

After testing my project, I have made the following evaluation:

Strengths:

- The chatbot replies very quickly – fast lookup
- Semantic Similarity works well
- IRC Server provides ease in communication
- Use of pickle files increases efficiency
- Storing user model data makes the chat personable

Weaknesses:

- Limited knowledge base
- User messages that are not similar to knowledge base can provide incorrect responses
- The chatbot needs to be running on my local machine

All in all, I believe that the biggest weakness was the limited knowledge base. If I had a larger, more extensive knowledge base, the chatbot would be able to handle many types of questions and answer correctly. My chatbot did work quite fast on my machine and I believe that semantic similarity parsing was the way to go!