Ryan Dimaranan (rtd180003)
Hannah Valena (hcv180000)

Chatbot Report and Evaluation

Instructions for Use

- View our chatbot here: https://replit.com/@hcvalena/ChatBot4395?v=1#main.py.
- Click the "Run" button to start the chatbot.

System Description

- To build this chatbot we used chatterbot, which is a python library that provides methods for training a chatbot on data such as corpuses and specific conversations.
- To get the training data, we used a variety of NLP techniques (listed below) and data retrieval methods such as web crawling and web scraping.
- To determine a response, the chatbot uses sentence similarity on the training data to get the best response. To break ties, the chatbot uses a randomizer to determine what response to output to the user.
- To save user personalizations, we use sentence parsing to determine when the user tells us something that they like or dislike, then store those terms into a dict while also training the chatbot with their specific likes and dislikes. After the user is done talking to the chatbot, this data is then saved into a pickle file to be used on the next run of the chatbot. As part of the start-up process the chat bot is trained on the user data, which allows for personalization for each user. The user state is determined by the name that the user inputs into our chat bot at first launch.

NLP Techniques

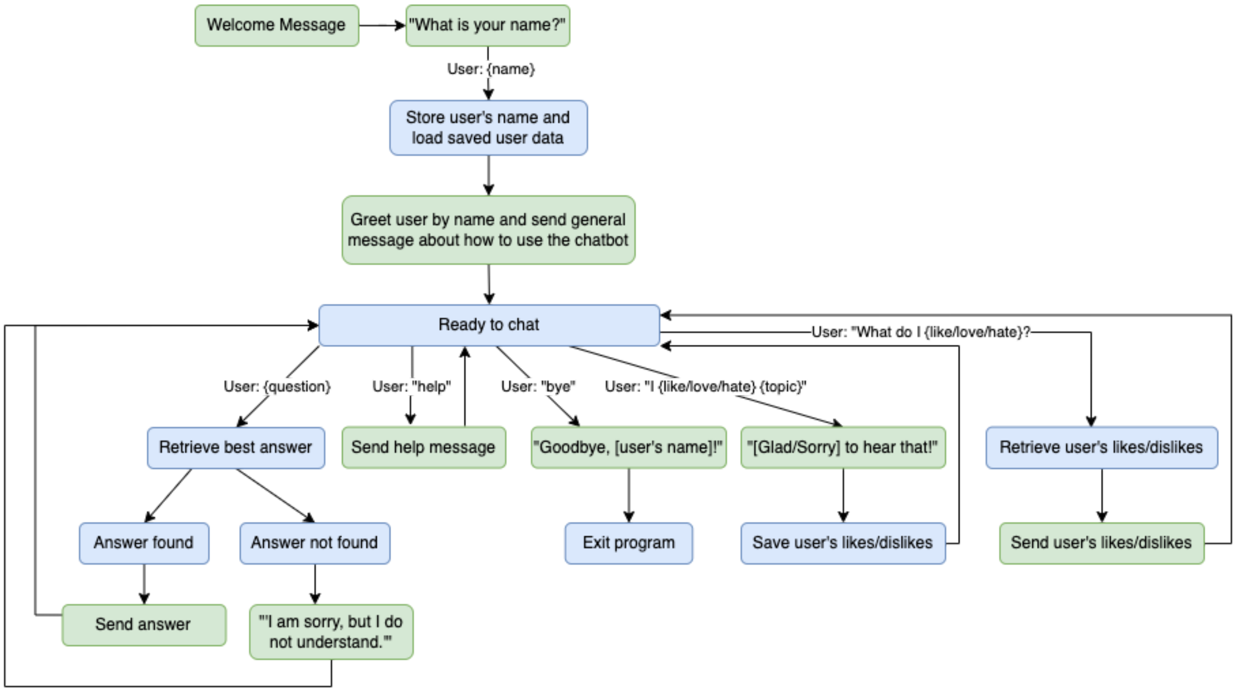
- Dependency parsing - We used a dependency parser on sentences to determine what the subject of the sentence was and the description or definition of the subject based on tags of other tokens or position of the word 'is'. We then used this information to train the chatbot to answer questions about definitions or facts such as "What is [term]?"

Ryan Dimaranan (rtd180003)
Hannah Valena (hcv180000)

Chatbot Report and Evaluation

- Named Entity Recognition (person) - We use named entity recognition to grab facts or actions about a specific person by getting the entity with a person tag, then training the chatbot to answer questions about what a specific person did or their significance to the topic.
- Named Entity Recognition (date) - We also use named entity recognition to detect a date in a sentence, extract that date and the event that happened, then train the bot to answer questions about events that happened on or around a specific date.
- TF-IDF for importance measure - Building upon the web crawler project, we use the same web crawler instead with max 50 sites to crawl and 40 top terms to increase our training data. Our web crawler uses urllib and BeautifulSoup to crawl and scrape web data. Then we clean the data by removing artifacts from HTML and removing stop words. We then convert the data to a TF-IDF dictionary which we then get the top 40 terms from the entire corpus of data. After getting these terms, we then go over the entire corpus again to retrieve related sentences to the terms we created which we then use as our knowledge base.

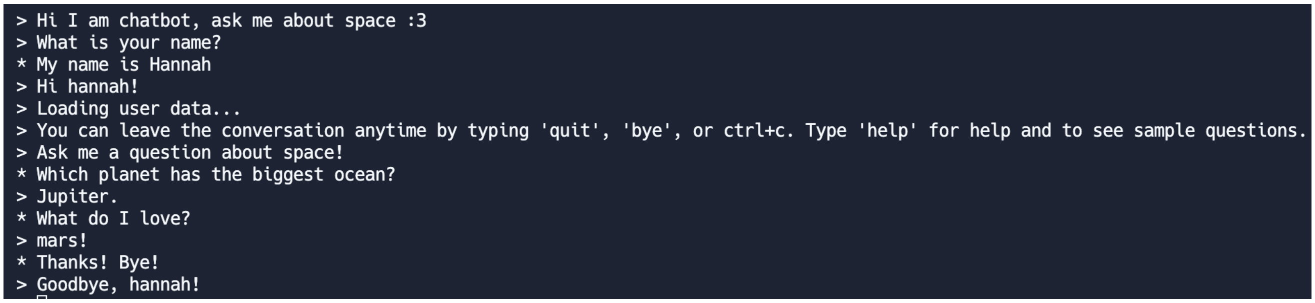
User Conversation Diagram/Logic


Welcome Message
"What is your name?"
User: {name}
Store user's name and load saved user data
Greet user by name and send general message about how to use the chatbot
Ready to chat
User: "What do I {like/love/hate}?
User: {question}
User: "help"
User: "bye"
User: "I {like/love/hate} {topic}"
Retrieve best answer
Send help message
"Goodbye, [user's name]!"
"[Glad/Sorry] to hear that!"
Retrieve user's likes/dislikes
Answer found
Answer not found
Exit program
Save user's likes/dislikes
Send user's likes/dislikes
Send answer
"'I am sorry, but I do not understand.'"

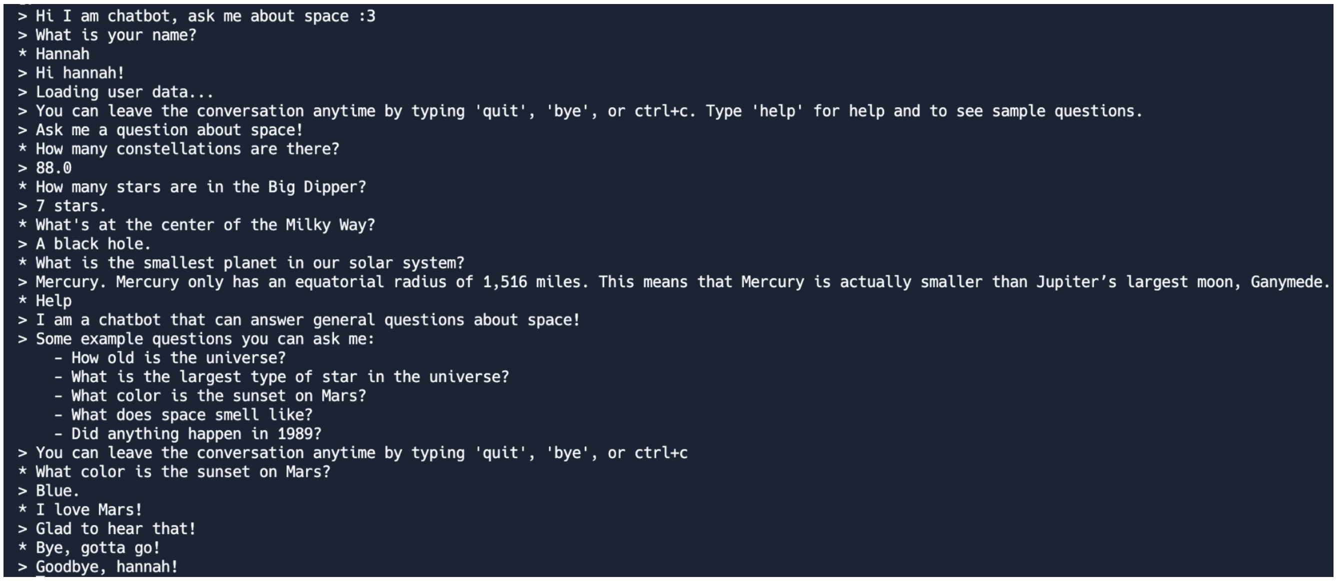
Ryan Dimaranan (rtd180003)
Hannah Valena (hcv180000)

Chatbot Report and Evaluation

Sample Dialogues



```
> Hi I am chatbot, ask me about space :3
> What is your name?
* My name is Hannah
> Hi hannah!
> Loading user data...
> You can leave the conversation anytime by typing 'quit', 'bye', or ctrl+c. Type 'help' for help and to see sample questions.
> Ask me a question about space!
* Which planet has the biggest ocean?
> Jupiter.
* What do I love?
> mars!
* Thanks! Bye!
> Goodbye, hannah!
```



```
> Hi I am chatbot, ask me about space :3
> What is your name?
* Hannah
> Hi hannah!
> Loading user data...
> You can leave the conversation anytime by typing 'quit', 'bye', or ctrl+c. Type 'help' for help and to see sample questions.
> Ask me a question about space!
* How many constellations are there?
> 88.0
* How many stars are in the Big Dipper?
> 7 stars.
* What's at the center of the Milky Way?
> A black hole.
* What is the smallest planet in our solar system?
> Mercury. Mercury only has an equatorial radius of 1,516 miles. This means that Mercury is actually smaller than Jupiter’s largest moon, Ganymede.
* Help
> I am a chatbot that can answer general questions about space!
> Some example questions you can ask me:
    - How old is the universe?
    - What is the largest type of star in the universe?
    - What color is the sunset on Mars?
    - What does space smell like?
    - Did anything happen in 1989?
> You can leave the conversation anytime by typing 'quit', 'bye', or ctrl+c
* What color is the sunset on Mars?
> Blue.
* I love Mars!
> Glad to hear that!
* Bye, gotta go!
> Goodbye, hannah!
```

Knowledge Base

We use 3 sources for our knowledge base:

- Original web crawler project
    - To create a knowledge base big enough for this project, we expanded our previous web crawler assignment to include data from 50 top terms from each site and crawl up to 50 sites relating to our topic. We also changed the starter URL that we previously used to one that included more factual data about space so that it is easier to parse and the information given is more reliable.

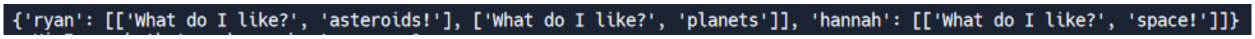
Ryan Dimaranan (rtd180003)
Hannah Valena (hcv180000)

Chatbot Report and Evaluation

- Separate web scraper of a space trivia site
    - To expand the knowledge base created from our original web crawler even more, we used BeautifulSoup to scrape 3 websites specifically about space trivia. These trivia websites contained lists of question-answer pairs, so we could train the chatbot on more specific questions about space. We cleaned and converted the scraped text into yaml and json files which are used as additional corpus training data.
- Saved user inputs
    - Chatterbot is able to have specific chatbot states for different users by use of a unique name for the bot. Using this we are able to grab the user's name at the start of the chatbot and load their specific chatbot with training data specific to that user. We store this data in a dictionary then use it to train the chatterbot for a personalized experience for each user.

User Models

- We use a dictionary that holds the person's name as a key (we retrieve when the chatbot first starts) and an array of trained conversations as its value. After the chatbot asks for the user's name, the bot then trains itself on the data provided by this dictionary to remember past responses that contain the user's likes or dislikes. The user model dictionary we use is found below.



```
{'ryan': [['What do I like?', 'asteroids!'], ['What do I like?', 'planets']], 'hannah': [['What do I like?', 'space!']]}
```

Chatbot Weaknesses

- This chatbot is heavily reliant on the training data that it is given, so the best quality training data must be used for best results.

Ryan Dimaranan (rtd180003)
Hannah Valena (hcv180000)

Chatbot Report and Evaluation

- This chatbot does not store complex information about users. Currently, only a user's name, likes, dislikes, and "hates" are stored.

Chatbot Strengths

- The response time of our chatbot is now typically just a few seconds, which allows for a more natural conversation to occur. Before optimization, the response time of our chatbot was around 60 seconds.
- Our chatbot is straightforward to use, since users don't have to download any files or software. They can access the chatbot on replit, an online IDE.

Chatbot Survey

- We created a Likert-style (1-5) survey to evaluate our chatbot. Our survey had 3 statements that users responded to on a scale of 1 (strongly disagree) to 5 (strongly agree):
    a. The chatbot was easy to use.
    b. The chatbot responded quickly to messages.
    c. The chatbot's responses were accurate.
- We found that 100% of those surveyed strongly agreed with the statement, "The chatbot was easy to use," and most people agreed with the statement, "The chatbot responded quickly to messages." However, most people surveyed disagreed with the statement, "The chatbot's responses were accurate." According to these results, our chatbot excelled with task success and efficiency, but performed poorly in terms of quality.

Ryan Dimaranan (rtd180003)
Hannah Valena (hcv180000)

Chatbot Report and Evaluation

Future Work

- The accuracy of our chatbot's responses can be increased by improving the quality of our knowledge base and training data. This can be done through further filtering to keep only the highest quality and most relevant data in our training data.
- The accuracy can also be improved by training our chatbot to be able to respond to many variations of the same question, which could be done with semantic parsing and expanding the training data to include more syntax variations.
- We could create a personality profile for our chatbot, so that a user could ask questions like, "What's your favorite planet?" or "What do you like about space?"