

Gebresilassie Takele
Biruk Mamo

CS 4395.001

Chatbot Project Report

System Description:

I made the decision to use the task-oriented dialogue agent chatbot instead of the more conversational chatbot out of the two types of chatbots we might create for this project. My chatbot is specifically made to respond to queries concerning the 1993 film *The Nightmare Before Christmas*. Possible responses that the chatbot might provide to the user are listed in the knowledge base, which is called `knowledge_base.txt` (see Appendix A). I carefully entered each and every statement in the knowledge base.

When a user asks my chatbot questions such as, "Who are the main characters in the movie?" or "How much money did the movie make?" or "When was the film released?", it performs well. You can see some examples of dialogue interactions in the report's later sections.

I used various common NLP processing methods for this project, such as tokenization, lemmatization, removing stop words and punctuation, etc. These common NLP processing methods were applied to the user input (i.e., whatever the user typed in) in order to (1) gather the most crucial information about the question and (2) match a knowledge base answer that most closely matches the question. Lines 49–73 of this code are the section that makes use of these methods.

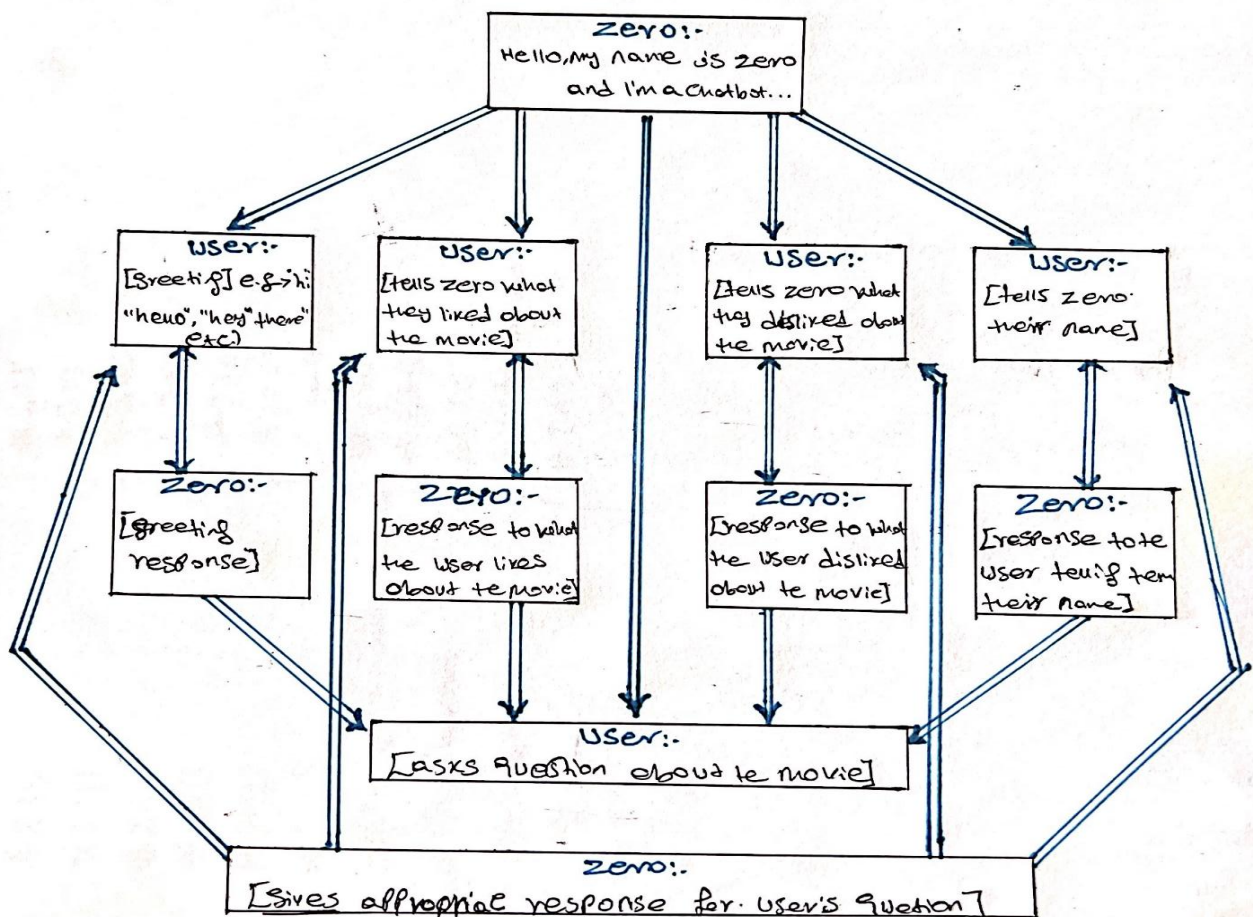
Aside from those NLP processing techniques, I also made use of the Tf-idf vectorizer and the cosine similarity metric. Utilizing these features was really the “meat” of creating my chatbot’s response. The Tf-idf vectorizer was used to get the vocabulary and the idf of the corpus. In this case the corpus is `sentence_tokens`, which contains the input from the user. The idf, or inverse document frequency, gives us a measurement of importance or relevance of a term in a given corpus. The cosine similarity measures how close two documents are. In this case, it’d be the user’s input compared to the knowledge base. Finally, I flattened and sorted the cosine values and got the highest one. If this value, which I called `highest_tfidf_value` *was not* 0, then that means there was an appropriate response found within the knowledge base. If this value, which I referred to as `highest_tfidf_value`, was not 0, it meant that the knowledge base included a suitable response. That response would then be displayed by the chatbot. However, if this number was 0, it meant that the chatbot didn’t find any suitable answers

in the knowledge base. If that were the situation, the chatbot's response would be a simple "Sorry, try again" message. The `chatbot_response()` method, which begins on line 88, contains all of this information.

And while not an NLP technique, I did add another feature to my chatbot that responds with fun facts if the user requests one. If the user types in "fun fact" or requests one (e.g., "Can you give me a fun fact?", "Please give me a fun fact", "I'd like to know a fun fact", etc.), then the chatbot will respond with a fact about the film. These facts are also present in the knowledge base—at the very end.

The conception/implementation of the user model will be discussed in a later part of the report (see Appendix B).

Dialogue Tree Diagram/Logic:



```

Hi, my name is Zero and I'm a chatbot that has knowledge about the movie 'The Nightmare Before Christmas'!
Ask me a question about the movie and I can answer it for you! If you want a fun fact about the movie, just type in 'fun fact'!
To end conversation with me, just type 'bye' or 'goodbye'.
hi
Zero: hey there
what's the runtime of the film?
Zero: the runtime of the film is 76 minutes.
I know there are three kids in the film, but I forgot their names
Zero: the names of the kids are lock, shock, and barrel.
which one is lock
Zero: lock is the one with the devil costume.
which one is barrel
Zero: barrel is the one with the skeleton costume.
which one is shack
Zero: Sorry, I didn't understand you. Please try again.
which one is shock
Zero: shock is the one with the witch costume.
can you give me a fun fact?
Zero: Fun fact: The film's set had trap doors that allowed animators to pop up and animate from below.
Zero: fun fact: there were about 13 animators working on the film.
who wrote the song in the film?
Zero: danny elfman wrote songs for the film.
who directed the movie
Zero: even though many people think tim burton directed the movie, henry selick actually directed it!
who voices dr.finkelstein
Zero: chris sarandon voices jack skellington's regular voice, but danny elfman voices his singing voice.
goodbye
Thanks for talking to me.

```

```

Hi, my name is Zero and I'm a chatbot that has knowledge about the movie 'The Nightmare Before Christmas'!
Ask me a question about the movie and I can answer it for you! If you want a fun fact about the movie, just type in 'fun fact'!
To end conversation with me, just type 'bye' or 'goodbye'.
Hello
Zero: hello! I'm happy to have someone to talk to!
where does Mayor's live
Zero: the mayor does live in halloween town.
who's the leader of halloween town
Zero: jack is the leader of halloween town.
who voices jack
Zero: chris sarandon voices jack skellington's regular voice, but danny elfman voices his singing voice.
what is the runtime of the movie
Zero: the runtime of the movie is 76 minutes.
who are the main characters of the movie
Zero: the main characters of the movie are jack and sally.
what does jack call santa clous in the movie?
Zero: santa is freed from oogie boogie by jack at the end of the movie.
what does jack call santa claus in the movie?
Zero: jack does call santa claus "sandy claws".
how much money did the movie make?
Zero: the amount of money the movie made was $91.5 million.
who voices he mayer?
Zero: chris sarandon voices jack skellington's regular voice, but danny elfman voices his singing voice.
bye
Thanks for talking to me.
(base) ment4m@Mentesinots-MBP Assignment8 %

```

Evaluation & Analysis of Chatbot:

This chatbot was created as a task-oriented agent that responded to inquiries concerning the movie The Nightmare Before Christmas, as was previously mentioned in this report. This makes it less conversational by nature than some other chatbots. Although I've written a few chatbot responses that sound conversational, the chatbot isn't meant to engage in conversation. The chatbot functions best as a question-and-answer tool, as you can see from the example dialogue interactions above. Again, it's not necessarily built for discussion. You can ask the chatbot for trivia

or express your feelings about the film, both of which should elicit appropriate responses.

When a user asks a question, the chatbot can respond, and I believe that most of the time, it does an excellent job of responding. The example dialogue interactions show that the chatbot is adept at responding to straightforward questions regarding the movie. Another advantage of this chatbot is that, in my opinion, it can respond to a wide range of queries about the movie. Since I totally constructed the knowledge base for this chatbot, I made an effort to provide as many details about the movie as I could.

You might inquire about the movie's financial success, its premiere date, the narrators of particular characters, etc. If you examine inside the knowledge base text file, you might also see that many of the entries are different versions of the same fact. I did this because the chatbot may respond to the user's question differently based on how they asked it. The user should receive the following response if they question, for instance, "How much money did the movie make?" The answer should be, "The amount of money the movie made was \$91.5 million." However, the chatbot would respond by using the term "film" rather than "movie" if the user had asked, "How much money did the film make?"

Additionally, the chatbot, as was already described in the study, employs cosine similarity and tfidf vectorization to identify important words and phrases in the user's input and then answers with a statement from the knowledge base that incorporates some of those words and phrases.

Having said that, one drawback of the chatbot and its capacity to generate responses is that sometimes your inquiries must be phrased in a particular way to receive the best response. For instance, you'll receive the response "the main characters of the movie are jack and sally" when you type in the inquiry "who are the supporting characters from the movie? ", which obviously does not address the issue. To avoid this, you should probably ask, "Who is the supporting cast of the movie?" in which case you'll hear, "The mayor, Dr. Finkelstein, and Oogie Boogie are the supporting cast of the movie."

Appendix A

Knowledge Base

The responses that Zero (the chatbot) might be able to come up with make up the knowledge base I've developed. I carefully entered each entry in the knowledge base that provides a response to a query regarding the film The Nightmare Before Christmas.

Depending on what the user asked, one of these responses (plus any of the other entries in the knowledge base) could be outputted by the chatbot.

Appendix B

User Models

The two factors on which my user models are based are the user's name and likes and dislikes. I believed that the only information that truly appeared relevant to store were the user's name and their thoughts about the film because this was supposed to be a task-oriented chatbot, especially with knowledge about The Nightmare Before Christmas.

You can view three user models produced by the chatbot by clicking the link in the sentence above. The dotted line separates each user model. As I previously mentioned, I only considered the user's name and their movie-related comments to be worthwhile to save.

You must express some things in a specific way if you want to test this out for yourself. For instance, if you want the chatbot to save your name, you must say "my name is". Similarly, if you want your comments about the movie to be saved, the chatbot will pick up on particular phrases (such as "I liked," "My Favorite," "I Thought," "I Disliked," "I Enjoyed," "I Wasn't Fan of," etc.). Starting at line 16 in the code, you can see more clearly which words and phrases the chatbot will answer to.