

Jesse Truong

Professor Mazidi

CS 4395.001

13 November 2022

## Chatbot Project

### **System Description:**

This chatbot was created to tell jokes based on what kind of jokes the user would like to hear. This chatbot is a rule based bot, meaning that the bot takes a specific keyword from the user input and assumes a conversation path of those keywords. Chat is the script for the chatbot

The chatbot joke database is based of a joke database on kaggle more specifically

(<https://www.kaggle.com/datasets/abhinavmoudgil95/short-jokes>)

The kaggle database has around 250,000 records of pure short-joke, good or bad one, safe for work and non-safe for work ones (unfortunately). There are **ALOT** of out of pocket jokes. So 250,000 records are way too many jokes for our small chatbot project. To correctly evaluate the entire database, a sample was taken from the data, receiving every 8th joke in the dataset, making the chatbot database to have a total of 31,250 jokes, which should cover most of the jokes the user would have requested.

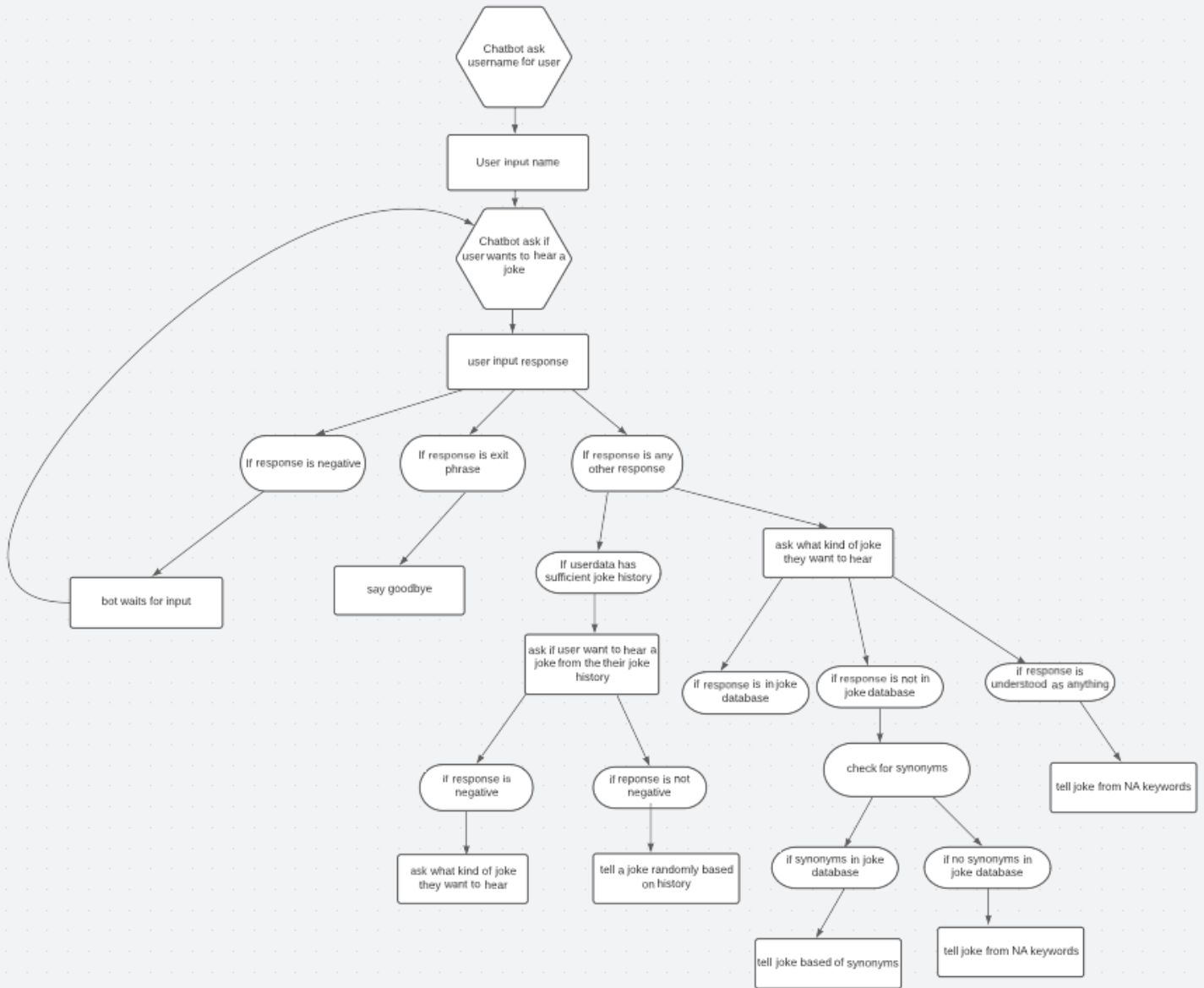
Within the data processing, correlating each joke to specific key words is crucial to the success of the bot. I couldn't figure out how to use POS tagging in python, but then figured that allenNLP could also do POS tagging, more specifically in the dependency parser. Every joke is parsed through the allen NLP dependency parser. All the words in the joke with the proper noun tag are put into a dictionary database with the proper noun as the key, and a list of strings which contain a joke. If no proper noun is found, we then check if nouns are found within the joke and that into the database. If no proper noun or noun are a joke, then the joke is still put into the joke database, but with a NA keyword to reference later. That type of joke would be used as a casual joke with no specific theme. After that all of the jokes have been processed, the dictionary is then pickled for the chatbot to use. The total process time is about 2 hours to go through the entire joke dataset.

The chatbot is first started by asking a username of the person, this is so we can receive or create a userdata base of which jokes to move toward or which jokes to avoid. After that, the chatbot ask if the user would like to hear a joke, the user can either exit the process or decline the jokes, which the chatbot would wait till the user input something. After that the chatbot would stem from two paths. One that would reference the user's joke history (if the history is large enough) and another would ask what kind of joke the user would like to hear. Every user response from this line is imputed through, again and allenNLP parser, tagging all of the words. The bot then checks all of the PropN and nouns in the input and checks if they match anything from the joke database. If no match is found. The bot uses the nltk wordnet to find synonyms from said proper nouns and nouns to again attempt to find a match. The joke is then said by the bot and a record of the joke keyword is then save locally to the chatbot

Most of the responses from the chat are generally unique, to give a human-like conversation, different sentences are used for the same function. For example, to ask if the user would want to hear a joke, there is a wide selection of dialog options the bot could select to chat with the user.

The user exits the chat on almost every dialog option the chatbot has. When the user exits, the joke history is then saved to the user database.

## Dialog Logic:



## Sample Dialog Interactions:

```
Jokebot: Hello there! I'm a humor chatbot trained on a Kaggle dataset, first off, what is your name?
Jesse
Jokebot: Greetings!
Jokebot: Would you like to hear a joke, Jesse?
Jesse: Sure
Jokebot: What kind of joke would you like to hear?
Jesse: Give me a Jackie Chen joke
Jokebot: Jackie Chan is in the new 'Star Wars' He's a Thai Fighter
Jokebot: I hope that was enjoyable.
```

```
Jokebot: Jesse, do you want to hear a joke?
Jesse: sure
Jokebot: A second joke about trump would you like to hear?
Jesse: That would be cool
Jokebot: Do you know why Trump's name is on all of his buildings? So the banks know which ones to take back! Ha cha cha!!
Jokebot: I hope you found that pleasant.
Jokebot: What type of joke are you interested in hearing?
Jesse: A joke about schwarzenegger
Jokebot: Did you know they are making an action movie about the great composers? Arnold Schwarzenegger says he'll be Bach.
Jokebot: I hope that was enjoyable.
```

```
Jokebot: What sort of humor are you looking for?
Jesse: anything
Jokebot: If you are reading this you are probably not blind.
Jokebot: I hope you found that pleasant.
Jokebot: Are you interested in a joke?
Jesse: Gimme a joke
Jokebot: What type of joke are you interested in hearing?
Jesse: money
Jokebot: How do you get money out of a piggy bank? Feed it laxatives.
Jokebot: I sincerely hope you enjoyed that.
```

```
Jokebot: A second joke about NA would you like to hear?
Jesse: sure
Jokebot: I love when you're choking someone and they are all "I can't breathe", duh I'm choking you.
Jokebot: I wish you had fun with that.
```

```
Jokebot: What type of joke are you interested in hearing?
robot123: What about a robot joke
Jokebot: What did the male robot say to the female robot? 1100011010000010001000100101011011110000101110100110100011001101011010000000110001001101011000001
Jokebot: I wish you had fun with that.
```

### **Appendix for the knowledge base:**

There are several text files that contain various dialog options of the chatbot for certain responses within the zip file. The text files are called JsecondQuestion.txt, altJquestion.txt, JQuestion.txt, endPharses.txt.

The post-processed jokes are in a pickle file called jokedatabase.p, but the pre-process data is in the csv file called shortjokes.csv. In the pickle file, joke database, the key is the word in the joke that had proper nouns or nouns and the value of the it is a list of string that contain said joke. The database build via the modelBuild.py script in the zip file

### **Appendix for the knowledge base:**

The userdata is stored in a pickle file called knowledgeBase.p, where the key is the username of the user and the value being a list of strings which contain the joke history. This is later reference to randomly ask if the user would like to hear a joke based on their joke history.

### **Evaluations of the Chatbot and Analysis of its Strengths and Weaknesses.**

Strength:

- Can store the joke history and refer back to the history when needed
- Large joke dataset for wide range of jokes
- If no jokes are found, a random joke can be selected
- Goodish jokes

Weakness:

- Terrible at long user response when asking for type of joke
- Can't understand the main point of user input.
- Alot of out of pocket jokes