

HLT – Fall 2020 – Chatbot Project

Charles Franznick

System description, including description of specific NLP techniques you used and how you used them:

JertBot is a chatbot that revolves around NLP techniques. It is not meant to simulate real human interaction; rather, it is meant to be as informative as possible. That being said, it is also meant to parse somewhat complicated sentences (“Tell me about Gandalf”) with ease.

The chatbot begins by loading the user database (a dictionary from a pickled file). It then asks the user’s name and uses this as the key for the dictionary, with the value being a list of topics the user has already asked about.

The program then has dialogue options for new and returning users respectively.

Then the bot takes user input (the question or topic) and begins the chat function. The chat function parses and tokenizes the user input using NLTK, removing stopwords and using POS tagging to determine the topic of interest from the user. The words that are important are then joined with underscores, as to be used in a wiki URL.

We then try the webpage to make sure it returns a webpage with content and not an error (i.e. 404). If this succeeds, then we reach another branch in the decision tree. If the user has looked up this topic before, the bot will reference that fact. If not, it will add to the user’s specific list of topics. It then calls the getInfo function to webscrape the lotr wiki and prints the output.

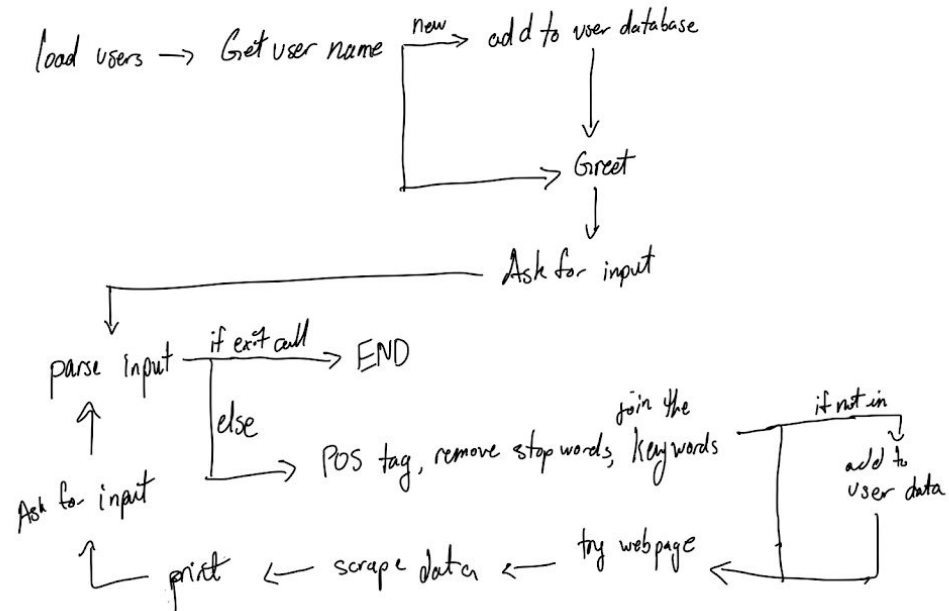
The getInfo function uses beautiful soup (bs) to get the html of the webpage, then uses bs to decompose (i.e. get rid of) a lot of stuff we don’t want (i.e. tables, license descriptions, etc.). Then we use a filter function to get only visible text, keep only the words before the word “Contents”, and clean the text using regex and NLTK’s sentence tokenizer. This clean text is returned to the chat function, where it is printed.

Built into this program, which can be enabled simply by calling ‘summarize’ (see line 95), is a function to summarize the text. This uses cosine difference between each vector (each sentence) and tf-idf to calculate the relative importance of each sentence and will output (default 2 sentences).

The final line of chat asks for user input once again, repeating the whole process unless the user inputs nothing ‘’ or one of the following keywords to end the program: 'bye', 'goodbye', 'see ya'

This program is easily expandable and it would be relatively straightforward to expand with external APIs.

Diagram of your dialog tree or logic:



Sample dialog interactions:

I am JertBot, a Lord of the rings info bot. What is your name? my name is charlie

Good to meet you Charlie. I know a lot about Lord of the Rings (but not J.R.R.Tolkien himself, yet).

You can tell me goodbye at any time and I will quit.

(Note: Right now I am only good at who and what questions. You can also try to just say the name of what you want to know about.)

What do you want to know about? *Gandalf*

Gandalf (Norse; IPA: - "Elf of the Wand" or "Wand-elf") the Grey, later known as Gandalf the White, and originally named Olórin (Quenya; IPA: - "Dreamer" or "Of Dreams"), was an Istar (Wizard), sent to Middle-earth in the Third Age to combat the threat of Sauron. He joined Thorin and his company to reclaim the Lonely Mountain from Smaug, convoked the Fellowship of the Ring to destroy the One Ring, and led the Free Peoples in the final campaign of the War of the Ring.

What do you want to know about? *Who is Frodo?*

Frodo Baggins, son of Drogo Baggins, was a Hobbit of the Shire during the Third Age. He was, and still is, Tolkien's most renowned character for his leading role in the Quest of the Ring, in which he bore the One Ring to Mount Doom, where it was destroyed. He was a Ring-bearer, best friend to his gardener, Samwise Gamgee, and one of the three Hobbits who sailed from Middle-earth to the Uttermost West at the end of the Third Age.

What do you want to know about? *bye*

Goodbye.

An appendix for the knowledge base (or live lookup) you created with samples

The bot will go to <https://lotr.fandom.com/wiki/> + some topic and find information from that webpage. For example, <https://lotr.fandom.com/wiki/Gandalf>. It then gets the first paragraph, ending at the beginning of the table of contents, and outputs that to the user.

An appendix for sample user models that were created

The user models were simple and based on any mention of a topic that the bot was successful in finding on the lotr fandom wiki. For example, the following user model contains 2 users: a default user just to make sure the bot loads the user base properly, especially when there is no pickle file to unpickle, and me, running the example dialogue above. User “Charlie” contains 2 searchers, Gandalf and Frodo.

```
{'default': 'default', 'Charlie': ['Gandalf', 'Frodo']}
```

Note that any user is saved in Title case, meaning any name will start with a capitalized letter and thus nobody can type default in - it becomes Default.

Evaluations of the chatbot and analysis of its strengths and weaknesses. Suggestion: you could get survey results from people who are not on the team using Likert-style questions, probably at least 3 questions

There is an obvious fault in the bot: if the user uses particularly weird syntax or searches for something not on the lotr wiki, it will fail. For example, Gandalf is fine but “Gandalf the Gray” searches https://lotr.fandom.com/wiki/Gandalf_Gray (due to removing stopwords) and thus returns nothing useful to the user. Otherwise, the bot does a good job of making accurate statements (due to the power of wikis) and summarizing the character or location or what have you nicely. The fault of this bot is of course large; implementing something external would improve performance, but I was unable to get the Google dialogflow working in time.