Figure HW7



Chatbot with NLP

We present - Ashley

Farzin Amiri, Trishal Varma

**Chatbot Report - Ashley**

Ashley is a chatbot that does a lot of teaching for chatbots in general, but the interesting thing is that it even does the getting to know of the user and showing the user it cares about the user’s likes and dislikes. It is so important for this chatbot to know the name of the user at every step of the way because the name is used as a key to the dictionary that it keeps the passions of the user and the knowledge base that the user is having. This chatbot takes things a step further and it also responds to the greetings of the user. The goal of all of this is to find a way to connect with the user while being most informative and attaining a good knowledge basis with the user. In my opinion, I believe this chatbot is very much so the superior than the chatbot use for googling things and this will have the last say when someone is trying to learn about chatbots. It is crucial to get to know about a good topic and a chatbot is the best way to do so because it just is. When trying to teach one pays more attention when the interaction is tailored and personalized.

The NLP techniques used are indeed very smart and come just in time. The Natural Language Processing happens once in the beginning and multiple times in the end and each time. The user enters a response and then the word lemmatizes, which leads to the response getting entered then parting and isolating the words in the response, which finally gets entered into the knowledge base. We can see the different Natural Language Techniques in depth when we dive deeper and look at the tokenizing of the statements. Here if the reader takes a closer look around line 45 it is decided that the words are not tokenized, and it is only the sentences that are taken apart. This is because the tdidf techniques would then later be used by the program to find similarity to the sentences. After tokenizing we do the tasks are lemmatizing and normalizing that to get it in a better shape and have it been more prone to training. Then we have a non nlp algo that just parses greeting statements and responds to common greetings that come up; this loop that does this task is intricate indeed. Then we run to the function of “respond to user” which of most importance and first figures out what type of input is provided to it by various types of investigation. This step is also of most importance because the user could be providing a greeting, personal detail, or just asking about the subject of chatbots!

If the user is requesting for a rather question, then by the way of tfidf a score is calculated and with cosine similarity we give it a similarity rating, if it’s high then that will be the basis for our response to the user. If the tfidf is 0 then the chatbot has nothing to say. This all gets done by the way of the getting the user’s name which is then queried in the likes and other knowledge base dictionary, and then by the way of specifying the knowledge base that is to be used we do the cosine similarity based on that.

The dialogue is quite simple, and the name is first asked to query the knowledge base is the corresponding dictionary, then Ashley asks about something the user likes or just anything about the user. This info is kept and used the next time the user makes a visit to the chatbot and the chatbot makes various tailored interactions based on that. And this is done because someone that is known personally can learn better than someone that really is disconnected from our chatbot Ashley. The process goes on and there is not a good result if the user doesn’t enter valid information about themselves. It is really down to the user to provide valid information, as everything that is said ends up in the hands of the chatbot and the experience will be more and more helpful.

Importing all the libraries that are to be used and will be used.

Text

Description automatically generated

Dictionaries that things are kept in, first one is for the knowledge base and the second one is for tailored interactions.

A picture containing graphical user interface, text

Description automatically generated

Here we are getting our knowledgebase from the webscraped text file:

Text

Description automatically generated

Checks if a certain key exists or not, and this function could be called when a certain key is to be entered and we don’t know if it is already in the knowledge base:

Text

Description automatically generated

Here we lemmatize and change the punctuation and normalize that lemmatization in there:

Text

Description automatically generated

We add a set of common greetings so we could attain appropriate responses, and the function “sayHiIt” does the rest by setting out what happens after saying those greetings.

Text

Description automatically generated

Here we respond to some one that is introducing themselves or adding their knowledge base into the dictionary and finding the tfidf values and such.

Text

Description automatically generated

The above we get the cosine similarity and the bottom we are getting the bot answer and such if the tfidf ends up being zero.

Before we initialize the chatbot, we also create a LDA model using Gensim package. This we use to understand the topics that are available in the web scraped document. We can use the commonly appearing topics to ask Ashley information about in the third question after letting her know what we like. The number of topics can be changed to a wider range to show how many different topics appear in the text which we preprocess. A screenshot of a cell phone screen with text

Description automatically generated

As shown, Topic 4 shows chatbots, whisky, Messi, and common words used.

Text

Description automatically generated

Text

Description automatically generated

Here we ask for name and get that particular knowledge base started then gain details about the user, and act accordingly to what type of detail is given thereafter.

Text

Description automatically generated

This here responds to the questions and the greetings and does it accordingly.

Text

Description automatically generated

Can leave the chatbot with response

**Dialog Tree**

Start

Ashley asks for Name of the User



Gives smart response from web scraped knowledge

Asks what they want to learn about?

Saves name, asks about what the user likes

Responds to I love you with a reply

Asks if they want to know more about a different topic

Responds to the User’s liking

Gives smart response from web scraped knowledge

**Sample Dialog Interaction**

Sample conversation with Ashley shows us that if we ask about Messi and Whisky, these are the responses she is able to give us from the knowledge base that is built by the web scraper. Since common topics included, Messi, Whisky, and Chatbot. These information are presented to the user when asked.

Text

Description automatically generated

**knowledge base**

Knowledge base that was made was with the scraping of the Wikipedia page about chatbots. These were combined into one document as a library that can be expanded to given information about multiple topics.

<https://en.wikipedia.org/wiki/chatbot>

<https://en.wikipedia.org/wiki/Messi>

<https://en.wikipedia.org/wiki/Whisky>

**Evaluation**

The chat bot is very strong in its early stages and can be expanded onto more topics in the future. The knowledge base is created on the web scraped documentation. The text documentation can have multitude of knowledge, or one specific to the user. This can be implemented in industry based environment with knowledge base for the company added into the documentation. The chatbot can recognize key words from the user input to find related answers in the documentation.

The improvement can be made on adding more functionality to the program with Machine learning which will allow the chatbot to learn more information and save training and test cases from previous users that have used it. This will allow the chatbot to make more smarter decisions when coming up with the response.