**LEBRON JAMES FAQ BOT REPORT**

INSTRUCTIONS

*To run on shell:*

1. Download zip folder and unzip to any preferred location, ensure all files are in the same folder.
2. Run “chat\_bot\_shell\_a3.py” and type “chat()” to run the chat function.
3. To prompt the bot to begin, type “hello”.
4. When asking a question, please ensure it ends with a “?”.
5. Refer to “questions.txt” to see what facts the bot knows about LeBron James and the questions you can ask it.
6. Type “goodbye” to end.

*To run on discord:*

1. Click on the following invitation link: <https://discord.com/api/oauth2/authorize?client_id=1062440289165389965&permissions=274877917184&scope=bot>
2. To prompt the bot to begin, type “hello”
3. When asking a question, please ensure it ends with a “?”.
4. Refer to “questions.txt” to see what facts the bot knows about LeBron James and the questions you can ask it.
5. Type “goodbye” to end.

VECTOR REPRESENTATIONS

I decided to use the TFidVectorizer over the CountVectorizer because I wanted to check not only the frequency of the word but the importance of it as well. The count vectorizer would match with more than one question in the question back too often and no matter what settings I changed it would still match with another question at least one time. When I changed to the TfidVectorizer with an ngram\_range(1,2) and included the stop words string I provided it, I was able to correctly match my intents even with a few typos.

Since the intent matching went so well with Tfid, I also used it to train my classifier as well. This portion of the assignment was to understand emotion and tone coming from an utterance and since a Tfid vectorizer gives a score on importance of words it worked better rather than just keeping a count by itself.

TRANSFER LEARNING

The way I incorporated deep learning was with word embedding vectors from spaCy to determine if a question is being asked. I first did this in the understand function in order to determine integer value I wanted to return from it because this integer would determine which type of response I wanted to make.

In the generate function I use the intent and the utterance as parameters and if there was a match detected from the intent in the understand function then it would simply return the corresponding index in the answer.txt file. However, if there was no match found, it then generates an answer to the question that was asked, if it was a question. This is where I used GPT3, with the settings set to max\_tokens=1024, n=1, stop=None, temperature=0.5 and the prompt set to the utterance. The utterance would determine what response will be generated by the API. If a question was not asked and it was just a statement, the bot then decides whether to reply to a statement that’s positive and a statement that’s negative respectively. This was done by pickling the documents from the sentiment folder.

THE PIPELINE

1. Utterance. I would take an utterance from a user and used lemmatization and a NLP to determine if a question was being asked or not.
2. Once it was verified that a question is being asked I moved onto intent matching using cosine similarity and Euclidean distance from the TfidVectorizer that I passed the intents through.
3. If the conditions weren’t met for a clear match to one of the questions, it now get passed through the GPT3 library to generate an appropriate response.
4. If there was no question and just a statement was made then it will go through our sentiment analyzation to determine if the statement was positive or negative, each will return the appropriate response.

THINGS TO TRY:

Here is a list of 11 things you can ask the bot:

1. What is the meaning of life?
2. How is the weather today?
3. Have you ever travelled to Europe?
4. Do you speak any other languages?
5. When did you realize you were built for greatness?
6. If you could fly to any country in the world, which would it be?
7. Do you love your job?
8. Do you hold yourself accountable when you are in the wrong?
9. How many planets are in the solar system?
10. I am having a great day.
11. I do not like you.