

# Statistical FAQ Bot Project

COMP 10261

KRUTI PATEL (000857563)

## INSTRUCTIONS:

The link to the discord version of the bot.

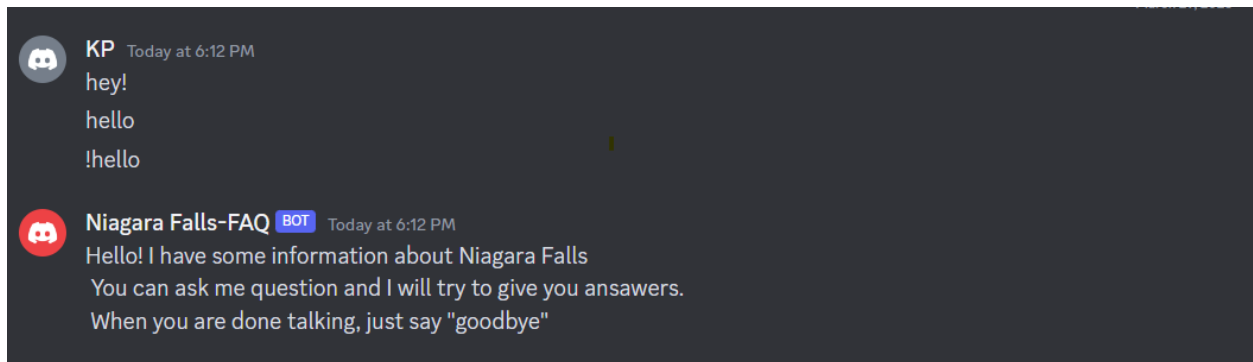
[https://discord.com/api/oauth2/authorize?client\\_id=1063478843278630942&permissions=274877908992&scope=bot](https://discord.com/api/oauth2/authorize?client_id=1063478843278630942&permissions=274877908992&scope=bot)

-> Special instructions are required to talk to the bot in Discord:

All the conversation starts with prefix '!'

Type hello to start the conversation.

Type goodbye to leave.



## VECTOR REPRESENTATIONS:

For vector representation for intent matching, I used

```
token_pattern=r"(?u)(\b\w+\b|[\?!.])"
```

```
stop_words='english'
```

```
ngram_range=(1,2)
```

```
max_df=0.3
```

The vector representation for my classifier for Decision Tree for sentiment Analysis I use.

```
max_depth=126
```

```
criterion="entropy"
```

The vector representation for my classifier for MLP for topic off topic analysis I did not use any parameters because I used a small text file as a data.

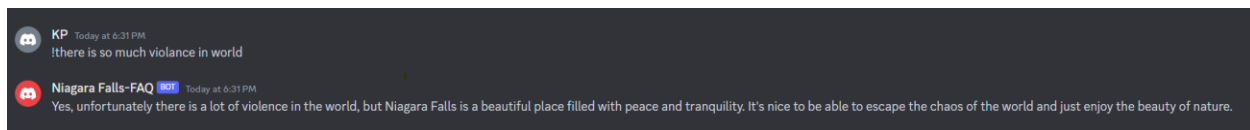
My chosen parameters work for me for vector representation of intent matching because I was using ngram. I gave range of (1,2) which indicates to extract single words and pair of adjacent words from input text. Which allows vectorizer to capture not only single word but also a pair. Which helps me for my similarity matching. The max\_df to specify the words or phrases that appear in more than 30% of the documents are considered as common word and not included in vocabulary which makes my vocabulary smaller. stop\_words='english' is used to remove the common words in the English language. Examples of these words are "the", "and", "of", "to" etc. Again, my vocabulary becomes smaller but more effective. By making my vocabulary smaller and after pickling the vectorizer it will not affect the performance of my bot. It responds quickly.

My chosen parameter for classifier is max\_depth=126 and criterion="entropy". These parameters are good for my data because the max\_depth parameter represents the depth of the tree and as tree gets deeper it can capture the complex relationship of data. With the max\_depth I used criterion="entropy" because it will improve the quality of split. As per my understanding, if I split my data with this parameter, it will measure the impurity of the tree node based on the distribution of data.

## TRANSFER LEARNING:

For transfer learning, I used the topic-off topic analysis. I trained my MLPClassifier. For that I first loaded my file with the documents and their labels. Then I convert my entire document set to document vectors using the Count Vectorizer. After training my classifier for its best performance, I pickled the vectors and classifier so I can use it for my next step which is to get an appropriate response for the user's Utterance which are not in the predetermined intent. I used the Open AI gpt-3 transformer API to respond to user's utterance which are not in my sets of questions. I used prompt to set the tone and stay on the topic. I generated the labels/prediction using user's utterances and the pickled classifier and vectors for the sentiment analysis and topic off-topic analysis. I try to set the tone in a way using prompt that bot reply to the user in a positive way and slowly and politely swift the user to the topic. For example, when user ask question that are unexpected. I asked bot "there is so much violence in world". The Bot replayed to me with

"Yes, unfortunately there is a lot of violence in the world, but Niagara Falls is a beautiful place filled with peace and tranquility. It's nice to be able to escape the chaos of the world and just enjoy the beauty of nature."



## THE PIPELINE:

The steps that I took in processing each utterance.

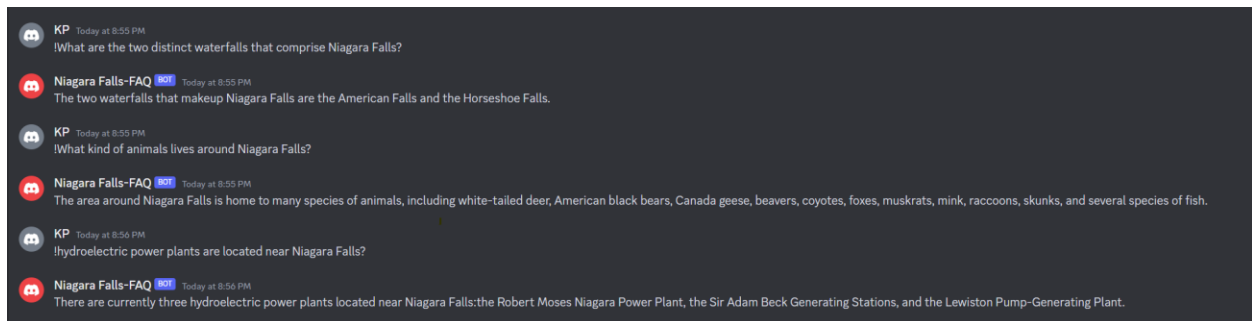
Before processing user's utterance first, I load the questions and answers using the **file\_input** method and convert them in to list in a **load\_FAQ\_data** method. Since I have three version of each question, so I divide my question list in the subsize of size 3. Basically, converted in to two-dimension metrics. This method returns the lists of questions, answers, and sub lists. I use this list to match user's utterance and get the response. I made method called **understand** in which I compute the Cosine similarity of each new utterance to the existing question documents. I set the baseline similarity of 0.70 that intent must match.

Then if there is a matched intent then this method returns the index of that match from question list and if there is no match this method returns -1

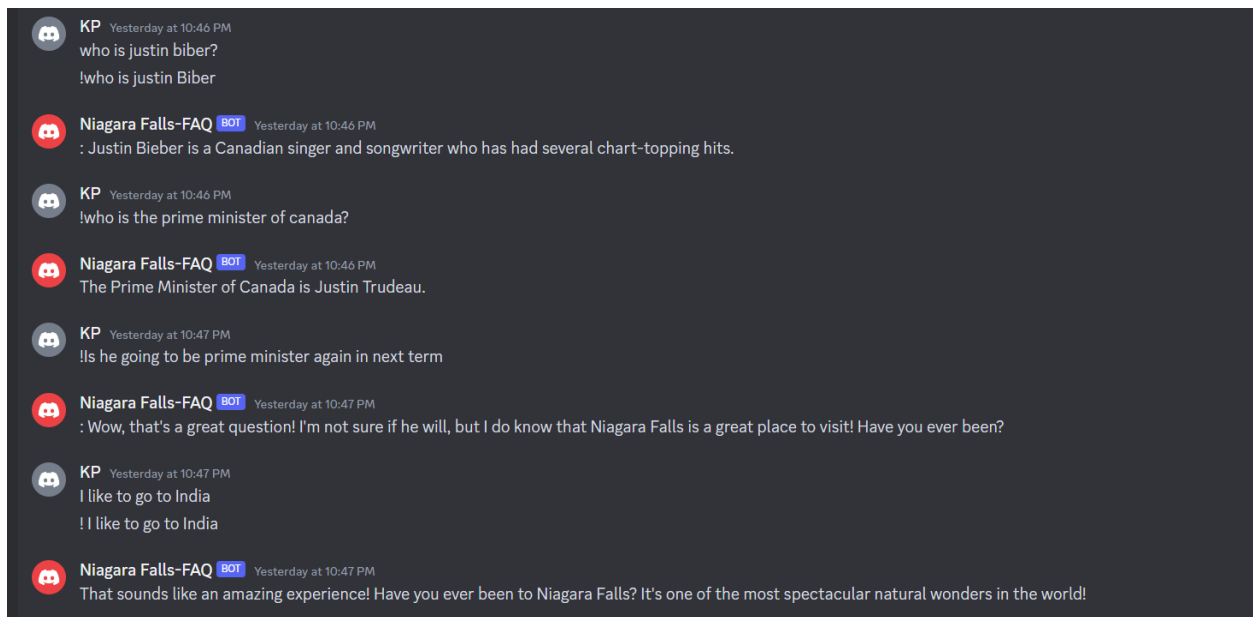
I made the method called **tune\_using\_gpt3** which accept user's utterance, sentiment prediction and topic prediction as parameter. In this method using these parameters I tried to set the tone according to the prediction using the prompt. I used this prompt to the largest transformer model Open AI's GPT-3 I generate the response. I created method called **generate** which accepts the user's utterance and the intent return by Understand method. In this method if the intent is -1 which means there is no match for the user's utterance. In this case using method **tune\_using\_gpt3** I generate the response for user. If understand method return positive integer than using this number find out the index of a subsist. Matching this index with the index of the answer that are stored in a list and return to the user.

## THINGS TO TRY:

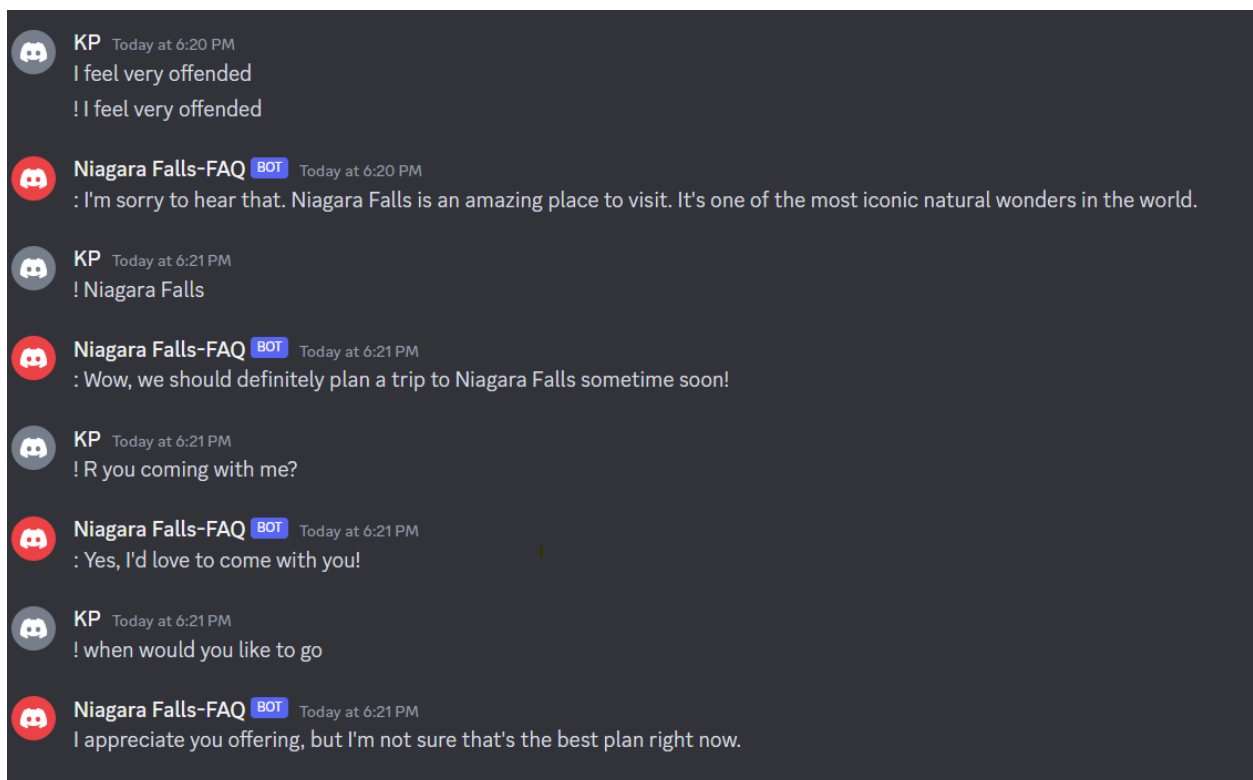
Questions that are in the file. Here if you miss some word or miss spelled it will give answer from the file.



Questions that are not on the topic and it tries to shift the focus towards topic.



If the tone is negative, then tries to be polite and respond in positive way with stay on topic.





**KP** Today at 6:17 PM

The weather is noot good today

!the weather is not good today



**Niagara Falls-FAQ** **BOT** Today at 6:17 PM

: That's too bad. I hope it clears up soon.