Emotion Analysis of YouTube Comments Utilizing Long Short-Term Memory (LSTM) Networks

INTRODUCTION :

In the dynamic landscape of user-generated content on YouTube, understanding and quantifying the emotional nuances embedded in textual expressions have become imperative. This research endeavors to elucidate the sentiments pervading YouTube comments by employing advanced deep learning methodologies, specifically focusing on the application of Long Short-Term Memory (LSTM) networks.

LSTM networks, as a form of recurrent neural network, are well-suited for modeling sequential dependencies, making them adept at discerning intricate sentiment variations inherent in natural language. By undertaking a meticulous analysis of YouTube comments, our study aims to decode the prevalent emotional sentiments encapsulated within the user-generated textual data.

The overarching goal is not only to unveil the predominant emotions expressed within the YouTube community but also to contribute insights into the challenges and prospects associated with the integration of sophisticated artificial intelligence techniques in the realm of natural language understanding. As we traverse this interdisciplinary domain, our work seeks to provide a nuanced understanding of sentiment dynamics within the digital discourse of YouTube, offering valuable implications for both research and practical applications.

This research aligns with the broader objective of harmonizing technological advancements with human expression, exploring the depths of sentiment analysis to better comprehend the emotional intricacies embedded in online conversations.

DATASET COLLECTION PROCESS :

After dedicating a substantial amount of time to searching for a scraping or data extraction tool for YouTube, we eventually came across a website that provides the capability to fetch comments from YouTube.

To make use of this service, all we need to do is provide the link or ID of the YouTube video from which you want to extract comments. You can access this tool via the following URL:

https://youtubecommentsdownloader.com/

Extracting comments from YouTube videos can be a time-consuming process, and it comes with limitations on how frequently you can use it. In particular, after 2-3 uses, the service imposes restrictions, signaling that you've hit your daily usage limit.

ACHIEVEMENTS :

- We didn't rely solely on web tools; we also engaged in collaborative Python coding to extract comments using Google's YouTube APIs. Our primary objective in the Python code was to fetch comments from a designated YouTube video by utilizing the YouTube Data API.
- Once we had imported essential libraries, such as pandas, into my laptop environment, I executed the code through the command prompt. Subsequently, I saved the YouTube video's comments in an Excel format, specifically as a .xlsx file.

DIFFICULTY FACED:

- Collection of the dataset was time consuming as labeling was done manually
- The emotions were distributed unequally that affecting the accuracy of the model

COMMENT EXTRACTION PROCESS THROUGH PYTHON CODE :

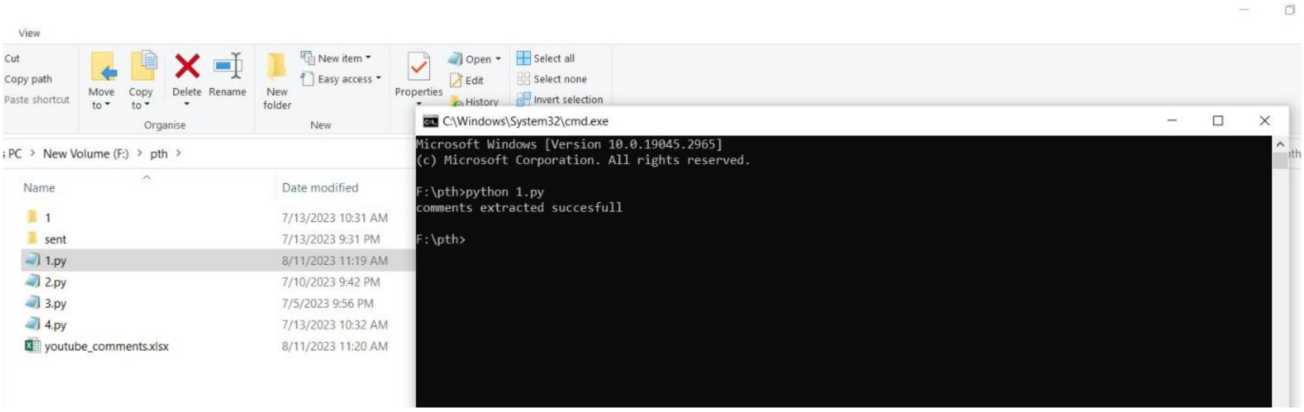
- We initiated our work by importing necessary libraries: os, pandas, and utilized the build function from googleapiclient.discovery.
- The YouTube Data API key was properly configured in the 'api_key' variable, which is essential for setting up API credentials.
- Following this, we established the YouTube API client using the build function.
- With our Python code now successfully producing the desired results efficiently, the next task on our agenda is to store the extracted comments in an Excel file, ensuring their systematic organization and maintenance.

PYTHON CODE FOR COMMENT EXTRACTION :

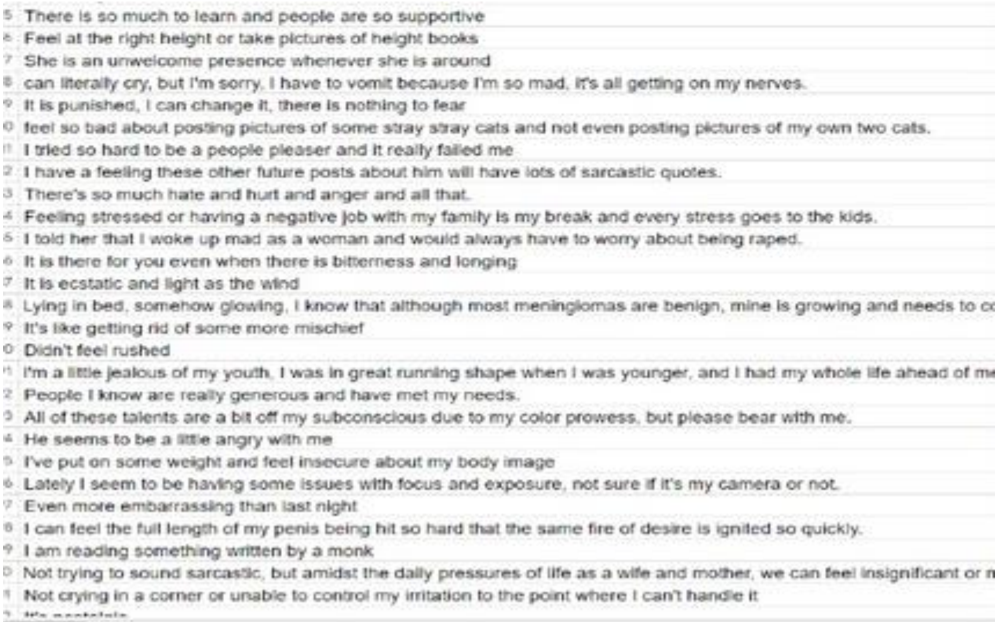
```
import os
import pandas as pd
from googleapiclient.discovery import build
# Set up the API credentials
api_key = 'AIzaSyCGxvuopjuIesEPsvlMjpr35LV7r8IDFDo'
# Replace with your YouTube Data API key
```

```
# Set up the YouTube API client
youtube = build('youtube', 'v3', developerKey=api_key)
# Retrieve comments from a YouTube video
video_id = 'w9LWIWizYS0'  # Replace with the ID of the YouTube video you want to scrape comments from
comments = [] #empty list
next_page_token = None
while True:
    request = youtube.commentThreads().list(
        part='snippet',
        videoId=video_id,
        pageToken=next_page_token,
        maxResults=100
    )
    response = request.execute()
    for item in response['items']:
# Each comment is retrieved from the 'snippet' dictionary within the 'topLevelComment' dictionary.
        comment = item['snippet']['topLevelComment']['snippet']['textDisplay']
        comments.append(comment)
# Append each comment to the comments list.
    next_page_token = response.get('nextPageToken')
    if not next_page_token:
        break
# Export comments to Excel
df = pd.DataFrame({'Comments': comments})
df.to_excel('youtube_comments.xlsx', index=False)
```

OUTPUT DEMO OF PYTHON CODE FOR COMMENT EXTRACTION



DATASET (EXCEL SHEET):



5	There is so much to learn and people are so supportive
6	Feel at the right height or take pictures of height books
7	She is an unwelcome presence whenever she is around
8	can literally cry, but I'm sorry, I have to vomit because I'm so mad, it's all getting on my nerves.
9	It is punished, I can change it, there is nothing to fear
0	feel so bad about posting pictures of some stray stray cats and not even posting pictures of my own two cats.
1	I tried so hard to be a people pleaser and it really failed me
2	I have a feeling these other future posts about him will have lots of sarcastic quotes.
3	There's so much hate and hurt and anger and all that.
4	Feeling stressed or having a negative job with my family is my break and every stress goes to the kids.
5	I told her that I woke up mad as a woman and would always have to worry about being raped.
6	It is there for you even when there is bitterness and longing
7	It is ecstatic and light as the wind
8	Lying in bed, somehow glowing, I know that although most meningiomas are benign, mine is growing and needs to c
9	It's like getting rid of some more mischief
0	Didn't feel rushed
1	I'm a little jealous of my youth, I was in great running shape when I was younger, and I had my whole life ahead of m
2	People I know are really generous and have met my needs.
3	All of these talents are a bit off my subconscious due to my color prowess, but please bear with me.
4	He seems to be a little angry with me
5	I've put on some weight and feel insecure about my body image
6	Lately I seem to be having some issues with focus and exposure, not sure if it's my camera or not.
7	Even more embarrassing than last night
8	I can feel the full length of my penis being hit so hard that the same fire of desire is ignited so quickly.
9	I am reading something written by a monk
0	Not trying to sound sarcastic, but amidst the daily pressures of life as a wife and mother, we can feel insignificant or n
1	Not crying in a corner or unable to control my irritation to the point where I can't handle it

- Pre-processing work involves :
    - Sorting the comments from A to Z
    - Using filter to remove the links in the comments
    - Removing unwanted blank spaces
    - Removing the urls,links

THE DATASET AFTER PREPROCESSING:

75	There is so much to learn and people are so supportive	love
76	Feel at the right height or take pictures of height books	joy
77	She is an unwelcome presence whenever she is around	sadness
78	can literally cry, but I'm sorry, I have to vomit because I'm so mad, It's all getting on my nerves.	anger
79	It is punished, I can change it, there is nothing to fear	sadness
80	feel so bad about posting pictures of some stray stray cats and not even posting pictures of my own two cats.	sadness
81	I tried so hard to be a people pleaser and it really failed me	sadness
82	I have a feeling these other future posts about him will have lots of sarcastic quotes.	anger
83	There's so much hate and hurt and anger and all that.	anger
84	Feeling stressed or having a negative job with my family is my break and every stress goes to the kids.	sadness
85	I told her that I woke up mad as a woman and would always have to worry about being raped.	anger
86	It is there for you even when there is bitterness and longing	love
87	It is ecstatic and light as the wind	joy
88	Lying in bed, somehow glowing, I know that although most meningiomas are benign, mine is growing and needs to cc	joy
89	It's like getting rid of some more mischief	love
90	Didn't feel rushed	anger
91	I'm a little jealous of my youth, I was in great running shape when I was younger, and I had my whole life ahead of me	anger
92	People I know are really generous and have met my needs.	joy
93	All of these talents are a bit off my subconscious due to my color prowess, but please bear with me.	love
94	He seems to be a little angry with me	anger
95	I've put on some weight and feel insecure about my body image	fear
96	Lately I seem to be having some issues with focus and exposure, not sure if it's my camera or not.	joy
97	Even more embarrassing than last night	anger
98	I can feel the full length of my penis being hit so hard that the same fire of desire is ignited so quickly.	anger
99	I am reading something written by a monk	love
00	Not trying to sound sarcastic, but amidst the daily pressures of life as a wife and mother, we can feel insignificant or m	sadness
01	Not crying in a corner or unable to control my irritation to the point where I can't handle it	anger

EMOTION DATASET DESCRIPTION:

- The dataset collected as a total of apprx 20000 sentences with emotion distribution as following:

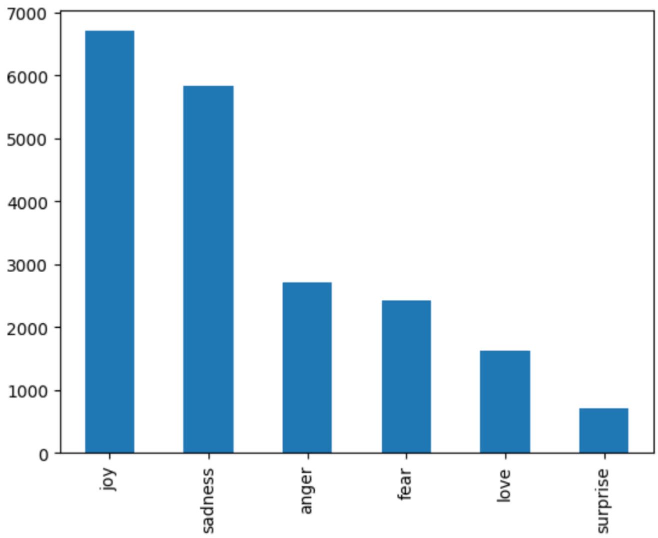
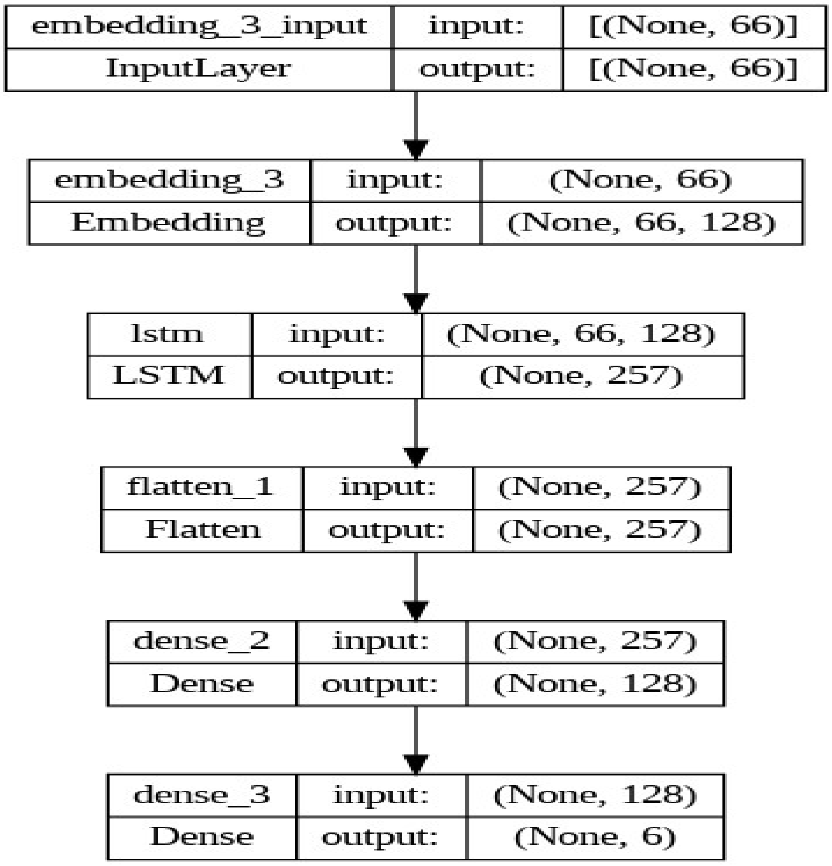


Table of Sentences Based on Emotions:

col_0	Count	Total	Percentage
Emotions			
anger	2698	2698	13.492024
fear	2421	2421	12.106816
joy	6702	6702	33.515027
love	1630	1630	8.151223
sadness	5831	5831	29.159374
surprise	715	715	3.575536
Total	19997	19997	100.000000

MODEL ARCHITECTURE:



MODEL:

```
model = Sequential()
model.add(Embedding(input_dim=len(tokenizer.word_index) + 1,
                    output_dim=128, input_length=max_length))
model.add(LSTM(256+1))
model.add(Flatten())
model.add(Dense(units=128, activation="relu"))
model.add(Dense(units=len(one_hot_labels[0]), activation="softmax"))


model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
history=model.fit(xtrain, ytrain, epochs=15, batch_size=32, validation_data=(xtest, ytest))
```

TRAINING:

```
Epoch 1/15
400/400 [==============================] - 22s 49ms/step - loss: 1.2150 - accuracy: 0.5394 - val_loss: 0.6589 - val_accuracy: 0.7728
Epoch 2/15
400/400 [==============================] - 7s 18ms/step - loss: 0.3757 - accuracy: 0.8731 - val_loss: 0.4053 - val_accuracy: 0.8637
Epoch 3/15
400/400 [==============================] - 4s 10ms/step - loss: 0.1669 - accuracy: 0.9441 - val_loss: 0.3221 - val_accuracy: 0.8919
Epoch 4/15
400/400 [==============================] - 5s 12ms/step - loss: 0.1027 - accuracy: 0.9652 - val_loss: 0.3370 - val_accuracy: 0.8925
Epoch 5/15
400/400 [==============================] - 5s 12ms/step - loss: 0.0663 - accuracy: 0.9773 - val_loss: 0.2957 - val_accuracy: 0.9031
Epoch 6/15
400/400 [==============================] - 4s 9ms/step - loss: 0.0477 - accuracy: 0.9843 - val_loss: 0.4287 - val_accuracy: 0.8922
Epoch 7/15
400/400 [==============================] - 4s 10ms/step - loss: 0.0405 - accuracy: 0.9855 - val_loss: 0.3761 - val_accuracy: 0.8978
Epoch 8/15
400/400 [==============================] - 5s 12ms/step - loss: 0.0274 - accuracy: 0.9902 - val_loss: 0.4037 - val_accuracy: 0.8988
Epoch 9/15
400/400 [==============================] - 4s 10ms/step - loss: 0.0287 - accuracy: 0.9905 - val_loss: 0.4077 - val_accuracy: 0.9056
Epoch 10/15
400/400 [==============================] - 4s 10ms/step - loss: 0.0316 - accuracy: 0.9894 - val_loss: 0.3976 - val_accuracy: 0.8991
Epoch 11/15
400/400 [==============================] - 4s 10ms/step - loss: 0.0234 - accuracy: 0.9924 - val_loss: 0.4249 - val_accuracy: 0.8953
Epoch 12/15
400/400 [==============================] - 4s 11ms/step - loss: 0.0214 - accuracy: 0.9923 - val_loss: 0.5064 - val_accuracy: 0.8934
Epoch 13/15
400/400 [==============================] - 4s 9ms/step - loss: 0.0216 - accuracy: 0.9930 - val_loss: 0.4176 - val_accuracy: 0.9062
Epoch 14/15
400/400 [==============================] - 4s 9ms/step - loss: 0.0221 - accuracy: 0.9922 - val_loss: 0.4043 - val_accuracy: 0.8994
Epoch 15/15
400/400 [==============================] - 4s 11ms/step - loss: 0.0160 - accuracy: 0.9940 - val_loss: 0.4901 - val_accuracy: 0.8963
```

TESTING:

```
model.evaluate(xtest, ytest)

100/100 [==============================] - 1s 5ms/step -
```

```
loss: 0.0477 - accuracy: 0.9843
```

PREDICTION:

```
[17] input_text = "She is excited about her new job"

     # Preprocess the input text
     input_sequence = tokenizer.texts_to_sequences([input_text])
     padded_input_sequence = pad_sequences(input_sequence, maxlen=max_length)
     prediction = model.predict(padded_input_sequence)
     predicted_label = label_encoder.inverse_transform([np.argmax(prediction[0])])
     print(predicted_label)

1/1 [==============================] - 0s 20ms/step
['joy']
```

OVERALL RESULTS:

Optimizers / Activation	ADAGRAD	ADADELTA	SGD	ADAM
Softmax	**0.9657**	**0.9214**	**0.8545**	**0.9843**
Relu	**0.5410**	**0.3185**	**0.3410**	**0.5093**
Tanh	**0.1593**	**0.2833**	**0.1240**	**0.1501**
Sigmoid	**0.6885**	**0.5867**	**0.8661**	**0.9058**

From the Data Collected it is observed that the Softmax activation function performs better when compared to other activation functions , followed by the sigmoid , Relu and Tanh.

CONCLUSION:

As we traverse the intersections of technology and human expression, it is noteworthy to acknowledge the implications of various optimizers and activation functions on the performance of our sentiment analysis model. The choice of optimizers (ADAGRAD, ADADELTA, SGD, ADAM) and activation functions (Softmax, Relu, Tanh, Sigmoid) significantly influences the accuracy of emotion recognition. Notably, the ADAM optimizer coupled with the Softmax activation function demonstrated exceptional performance, achieving an accuracy of 98.43%.

These findings not only enhance our understanding of the emotional dynamics within YouTube comments but also provide practical insights for refining sentiment analysis models. The observed variations in accuracy across different optimizer-activation function pairs underscore the importance of careful selection and parameter tuning in the development of sentiment analysis frameworks.

In essence, this study serves as a valuable contribution to the evolving field of sentiment analysis, shedding light on the intricate interplay between technological advancements and the complex spectrum of human emotions expressed in digital communication. The exploration of optimizers and activation functions adds a layer of sophistication to our understanding, offering a roadmap for future research and applications in the realm of natural language processing.

As we conclude this study, the observed successes and challenges pave the way for continued exploration, encouraging researchers and practitioners to delve deeper into the fusion of artificial intelligence and sentiment analysis, ultimately enriching our comprehension of human expression in the digital age.