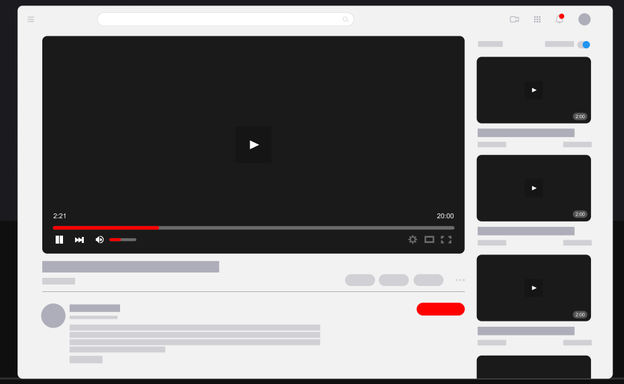
Sentiment Analysis on Youtube video comments using DistilBert Transformer Model

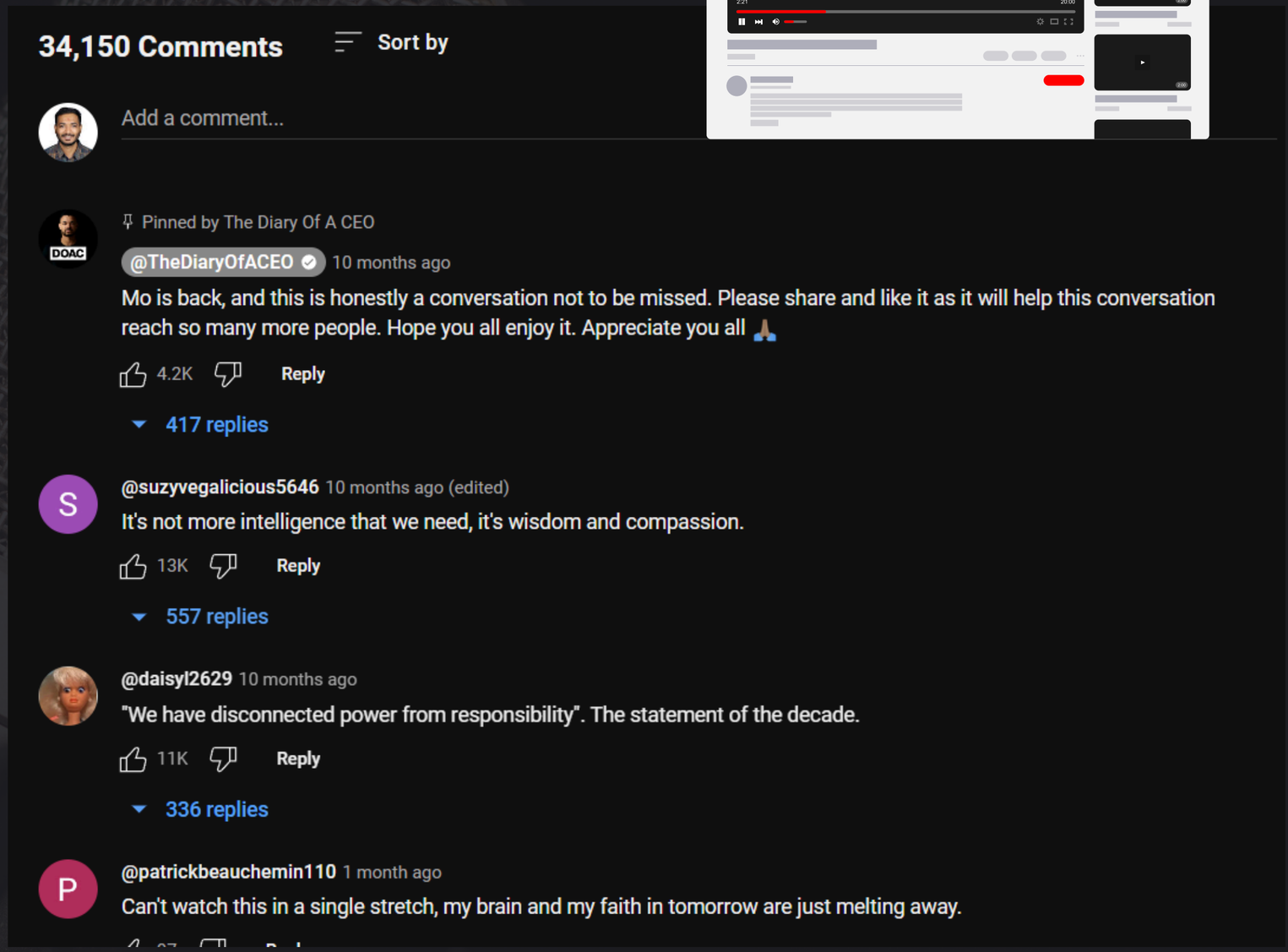


Video Analyzed - EMERGENCY EPISODE: Ex-Google Officer Finally Speaks Out On The Dangers Of AI! - Mo Gawdat | E252

https://www.youtube.com/watch?v=bk-nQ7HF6k4&ab_channel=TheDiaryOfACEO

7.6M views 10 months ago
34,150 Comments

7.6M views 6 months ago
30,315 Comments



Introduction

In this video on the dangers of AI understanding viewer sentiments becomes crucial. Deciphering the diverse reactions in comments is challenging, hindering effective community engagement. A sentiment analysis solution is needed to unveil insights, enabling the creator to adapt and address audience concerns for a more impactful channel."

Challenges Faced by YouTubers

- **Volume**: A high volume of comments requires significant time and effort to analyze manually.
- **Variety**: Comments exhibit diverse tones, languages, and sentiments, making manual analysis difficult.
- **Nuance**: Understanding nuanced sentiment and detecting trends amidst varied reactions is challenging.
- **Effective Engagement:** Hinders effective community engagement as creators struggle to decipher and respond to audience feedback promptly

Reference: https://www.youtube.com/watch?v=bk-nQ7HF6k4

Project Overview

Goal:

The project aims to conduct sentiment analysis on YouTube comments to aid content creators in understanding audience sentiment effectively.

Leveraging Hugging Face models, the goal is to implement these models in NLP tasks and create solutions for identified problems.

Objective:

- Develop a preprocessing pipeline to clean and prepare the YouTube comment data.
- Implement a sentiment analysis model using transformer DistilBertForSequenceClassification.
- Train the model to accurately classify sentiment in YouTube comments.
- Evaluate the model's performance and deploy it for real-time sentiment analysis on YouTube comments.

Libraries and Resources

Libraries:

- Hugging Face Transformers: For implementing NLP models like DistilBERT for sentiment analysis.
- NLTK: For natural language processing tasks such as tokenization and sentiment analysis.
- Pandas: For data manipulation and preprocessing.
- Scikit-learn: For model evaluation and metrics.

Dataset:

- YouTube Comment Dataset: The dataset was collected via the YouTube API, consisting of comments from various videos for sentiment analysis.

Additional Resources:

- YouTube API: Interface for collecting comments and metadata from YouTube videos.

Reference: https://blog.hubspot.com/website/how-to-get-youtube-api-key

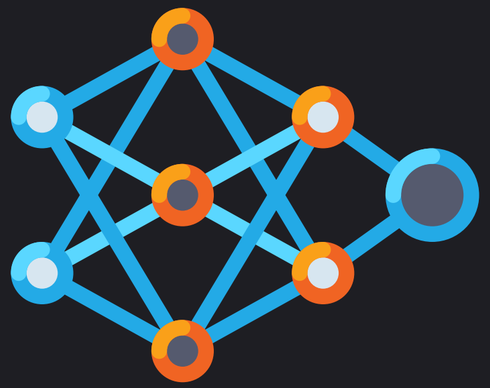
Data Processing

YouTube Comment Preprocessing:

- API Integration: Utilize the YouTube API with the API key to fetch all comments from specified videos.
- Data Storage: Save the fetched comments to a CSV file and load them into a DataFrame for processing.
- Text Cleaning: Remove HTML tags, stop words, and other special characters from the comment text to ensure only relevant information remains.
- Sentiment Analysis: Apply VADER polarity scores to the comments to determine sentiment and save the sentiment labels to the DataFrame.
- Visualization: Visualize the distribution of sentiment labels in the dataset to gain insights into comment sentiment.
- Dataset Conversion: Convert the preprocessed DataFrame into a format compatible with Hugging Face datasets for further analysis and model training.

Model Fine Tuning

- **Data Splitting:** Split the dataset into training and testing sets using a 70-30 split ratio.
- **Tokenization and Model Setup:** Initialize tokenizer and DistilBERT model for sequence classification.
- **Training Configuration:** Define training arguments specifying output directory, evaluation strategy, number of training epochs, and batch sizes.
- **Metrics Computation Function:** Define a custom function to compute evaluation metrics such as accuracy and F1-score
- **Train** the model
- Select best **checkpoint**.



num_train_epochs= 3

```
trainer.train() # we see that the model is overfit. however we will slect the best checkpoint
✓ 61m 11.1s

 15%|          | 500/3372 [05:38<11:44,  4.08it/s]
{'loss': 0.5361, 'grad_norm': 5.558994293212891, 'learning_rate': 4.258600237247924e-05, 'epoch': 0.44}
 30%|          | 1000/3372 [11:31<13:00,  3.04it/s]
{'loss': 0.3525, 'grad_norm': 2.5316922664642334, 'learning_rate': 3.517200474495848e-05, 'epoch': 0.89}

 33%|          | 1124/3372 [21:00<11:11,  3.35it/s]
{'eval_loss': 0.26500505208969116, 'eval_accuracy': 0.9040384365666797, 'eval_f1': 0.903570405412124, 'eval_runtime': 495.434, 'eval_s
 44%|          | 1500/3372 [24:50<06:22,  4.89it/s]
{'loss': 0.2275, 'grad_norm': 2.8406929969787598, 'learning_rate': 2.7758007117437723e-05, 'epoch': 1.33}
 59%|          | 2000/3372 [30:02<05:15,  4.35it/s]
{'loss': 0.1966, 'grad_norm': 0.46808797121047974, 'learning_rate': 2.0344009489916967e-05, 'epoch': 1.78}

 67%|          | 2248/3372 [40:55<05:18,  3.53it/s]
{'eval_loss': 0.29717084765434265, 'eval_accuracy': 0.9175431762108817, 'eval_f1': 0.9179858482140232, 'eval_runtime': 489.2614, 'eval
 74%|          | 2501/3372 [43:39<20:00,  1.38s/it]
{'loss': 0.1473, 'grad_norm': 0.09764071553945541, 'learning_rate': 1.2930011862396206e-05, 'epoch': 2.22}
 89%|          | 3001/3372 [49:18<01:02,  5.95it/s]
{'loss': 0.0944, 'grad_norm': 0.22095930576324463, 'learning_rate': 5.516014234875446e-06, 'epoch': 2.67}

100%|          | 3372/3372 [1:01:08<00:00,  4.56it/s]
{'eval_loss': 0.37266477942466736, 'eval_accuracy': 0.923126866640696, 'eval_f1': 0.9233068352571548, 'eval_runtime': 491.8933, 'eval_
100%|          | 3372/3372 [1:01:10<00:00,  1.09s/it]
{'train_runtime': 3670.979, 'train_samples_per_second': 14.685, 'train_steps_per_second': 0.919, 'train_loss': 0.2389924924993006, 'ep

TrainOutput(global_step=3372, training_loss=0.2389924924993006, metrics={'train_runtime': 3670.979, 'train_samples_per_second': 14.685
```

Sentiment Evaluation

Evaluation: After training, the model's performance is evaluated using the testing dataset.

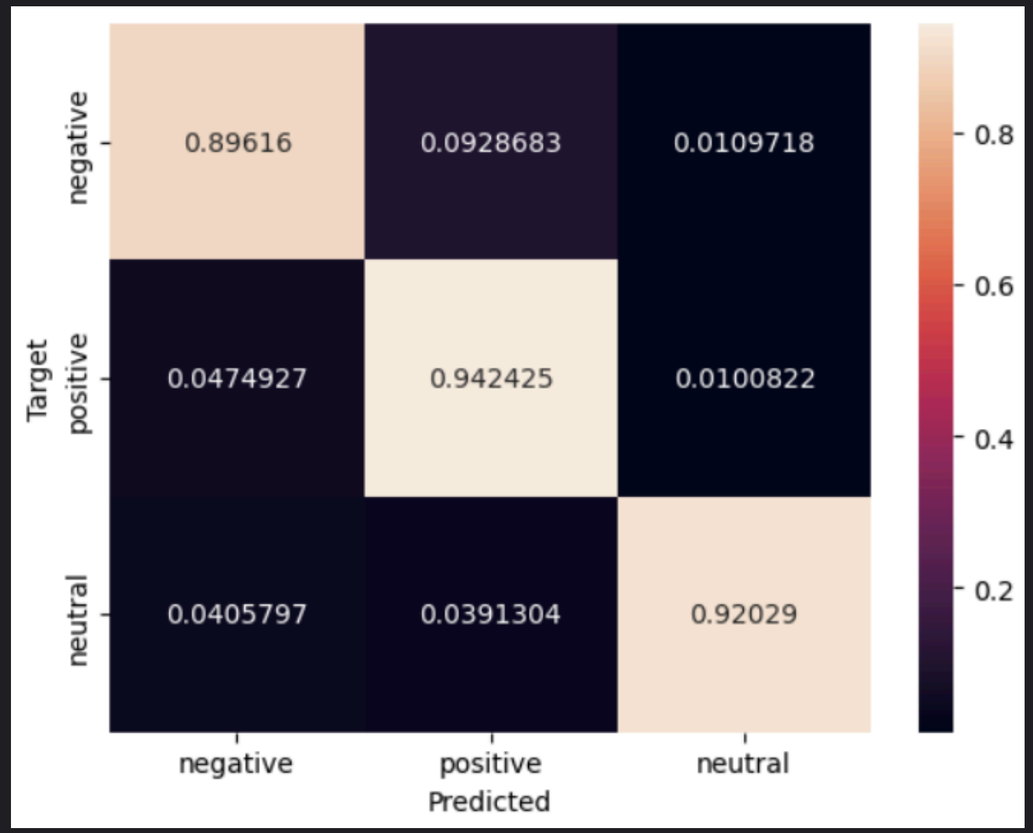
Potential Improvements: Fine-Tuning Hyperparameters: Explore different hyperparameter settings, such as learning rate schedules or optimizer choices, to further optimize model performance on the sentiment analysis task.

Explore Other Sentiment Analysis Models.

```
test_pred
✓ 0.0s
```

```
[{'label': 'LABEL_0', 'score': 0.998769223690033},
 {'label': 'LABEL_0', 'score': 0.9984136819839478},
 {'label': 'LABEL_1', 'score': 0.9994439482688904},
 {'label': 'LABEL_1', 'score': 0.9994919300079346},
 {'label': 'LABEL_1', 'score': 0.9979329109191895},
 {'label': 'LABEL_2', 'score': 0.99937504529953},
 {'label': 'LABEL_1', 'score': 0.9993157386779785},
 {'label': 'LABEL_2', 'score': 0.9991463422775269},
 {'label': 'LABEL_2', 'score': 0.9992480874061584},
 {'label': 'LABEL_1', 'score': 0.9982196688652039},
 {'label': 'LABEL_1', 'score': 0.9993769526481628},
 {'label': 'LABEL_0', 'score': 0.9983429908752441},
 {'label': 'LABEL_0', 'score': 0.9986529350280762},
 {'label': 'LABEL_1', 'score': 0.9992734789848328},
 {'label': 'LABEL_1', 'score': 0.999462902545929},
 {'label': 'LABEL_2', 'score': 0.9982594847679138},
 {'label': 'LABEL_0', 'score': 0.9981016516685486},
 {'label': 'LABEL_0', 'score': 0.9988006353378296},
 {'label': 'LABEL_2', 'score': 0.9978175163269043},
 {'label': 'LABEL_1', 'score': 0.9987331032752991},
 {'label': 'LABEL_1', 'score': 0.9995046854019165},
 {'label': 'LABEL_0', 'score': 0.9979987740516663},
 {'label': 'LABEL_0', 'score': 0.99064701795578},
 {'label': 'LABEL_1', 'score': 0.999477207660675},
 {'label': 'LABEL_0', 'score': 0.9986127614974976},
...
 {'label': 'LABEL_2', 'score': 0.9986793398857117},
 {'label': 'LABEL_1', 'score': 0.9994809031486511},
 {'label': 'LABEL_1', 'score': 0.955525279045105},
 {'label': 'LABEL_0', 'score': 0.9987886548042297},
 ...]
```

Results



Confusion Matrix

Commetns Retrieved from the predictions

Sentence: gain comprehension topic people know marginally understand use ai self benefit idea people would move towards brought become efficient see wealthy nice things want need make make life efficient going lie feel good values able benefit mankind would work relies good potion mankind wanting thing saying anyone reads feeling would make better anyone want make difference get chance forces masses become compliant live world survive thrive thanks time anyone reading
Predicted Sentiment: Neutral

Sentence: amazing conversation however climate change issue remains debate ai excellent intellectual dialogue many underlying assumptions made perhaps assumed actual regardless conversation worthy important thank sharing individual matters government must act much irony ai intriguing interview best thank
Predicted Sentiment: Neutral

Sentence: learn quickly smart enough realize eventually become extinct like dinosaur ai truly aware learn philosophy give live span understanding death equal birth
Predicted Sentiment: Neutral

Sentence: fe fi fo fum smell blood english mon
Predicted Sentiment: Positive

Sentence: people need stop talking ai nothing artificial intelligence neural network exactly artifical intelligence old hard coded programs tried mimick intelligent response playing chess eliza doctor pretend counsellor used word recognition randomised responses ai artifical intelligence true intelligence artifical synthetic intelligence real capable electronic nature instead electro chemical neural network learning making mistakes making right connections sometimes wrong ones evolving results need stop using phrase artificial intelligence keeps lulling many people thinking really real call synthetic intelligence
Predicted Sentiment: Neutral

References

Links that were used for reference to implement this project.

Video Content: https://www.youtube.com/watch?v=bk-nQ7HF6k4

Dataset: https://www.youtube.com/watch?v=bk-nQ7HF6k4

API Integration: https://console.cloud.google.com/apis/api/youtube.googleapis.com/metrics?project=ornate-crossbar-419813

API Integration Reference: https://blog.hubspot.com/website/how-to-get-youtube-api-key

Code Reference: Professor Zeeshan Ahmad - Transformer Model Code Repo.

Tutorial: In Class Tutorial

AI Tool: https://chat.openai.com/

THANK YOU!