SENIOR PROJECT CN3 – 1/2024

Multi-Channel Sentiment Analysis

Project Concept

Submitted to

School of Information, Computer and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University

September, 2024

By

Matas	Thanamee	6422771251
Piraboon	Piyawarapong	6422781466
Napat	Ariyapattanaporn	6422782399
Teetawat	Bussabarati	6422782423

Advisor: Dr. Cholwich Nattee

Project Overview

Abstract

This project introduces a sentiment analysis system that integrates three distinct data channels: facial expression, voice tone, and speech transcription. Each channel independently analyzes sentiment, classifying emotions as either positive or negative. Through preprocessing the outputs and applying a voting mechanism, the system provides an alternative, lightweight, and more explainable approach to sentiment analysis compared to traditional multimodal techniques, which rely on early fusion within neural networks. Our approach is designed to work alongside existing multimodal methods, offering customizable edge case detection, while allowing the use of multimodal systems for more standard cases. This combination enhances flexibility and transparency in human emotion analysis, allowing researchers to tailor solutions for specific needs.

Introduction

The ability to accurately detect and understand human emotions is a key factor in various technological applications, virtual avatars that dynamically adjust their facial expressions to match the user’s emotions, systems that monitor mental health, and solutions for enhancing customer service experiences.

Historically, sentiment analysis systems have primarily relied on a single data channel, such as textual data, to interpret and classify emotional states. However, human emotions are complex, and expressed through a combination of verbal and non-verbal cues. Traditional single-channel sentiment analysis may miss subtle emotional signals, failing to accurately reflect a person's true emotional state. For instance, someone might feel sad but mask it, or sound angry but be joking. Therefore, it is essential to consider not just the content of spoken words but also the tone of voice and facial expressions. Each of these channels provides distinct insights into emotional states:

1. Facial recognition detects visual cues that often convey emotions more immediately and universally than verbal expression.
2. Speech transcription provides semantic content, allowing for analysis of word

choice and linguistic patterns associated with different emotions.

3. Voice tone analysis captures vocal features that can reveal emotions not explicitly stated in words.

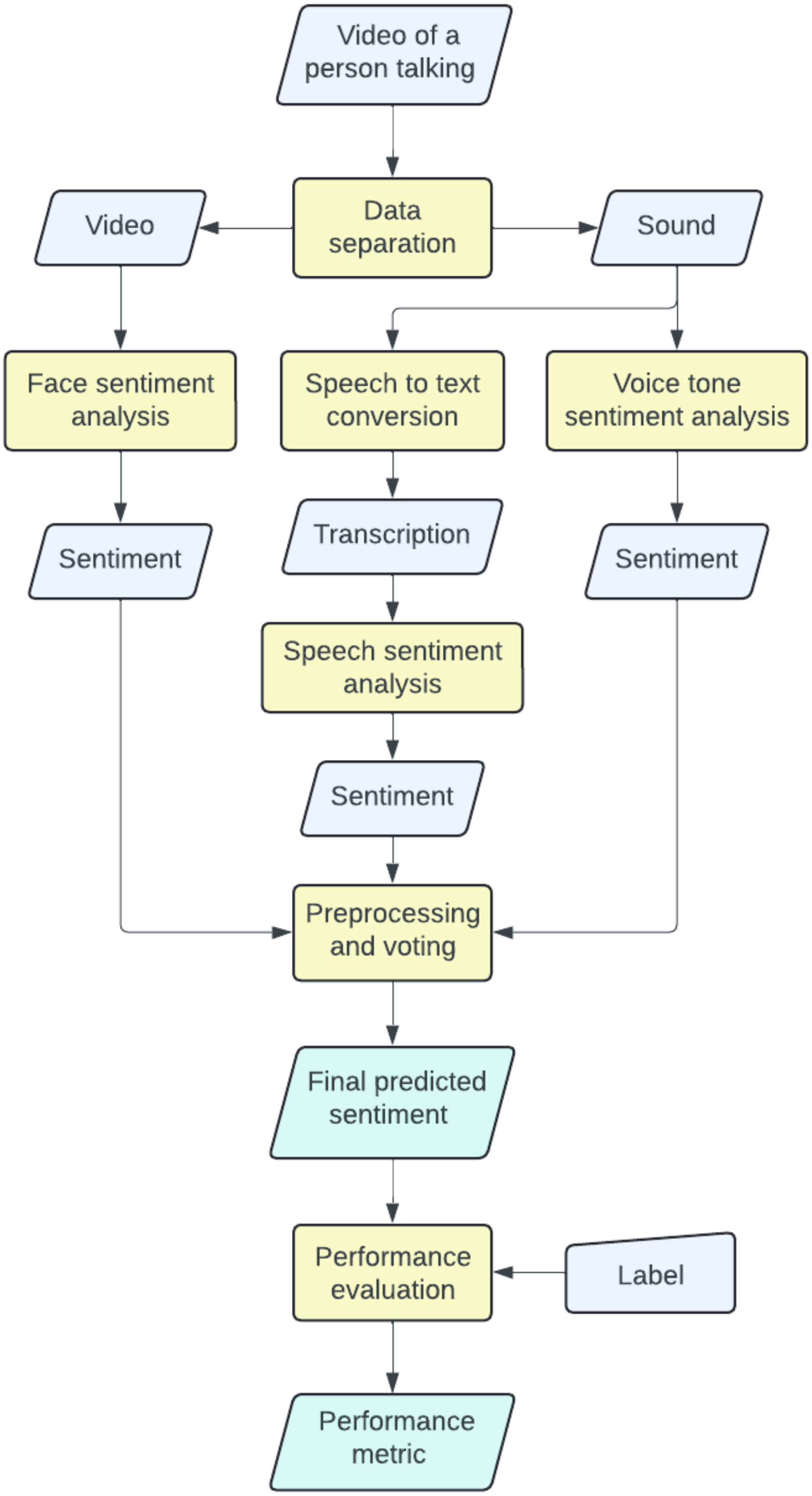
For example, while text-based systems may be able to interpret the meaning behind words, they might miss the nuances conveyed through the speaker's tone or facial expressions. As noted by Nandwani and Verma in their 2021 paper, *“A review on sentiment analysis and emotion detection from text”,* “… in some cases, machine learning models fail to extract some implicit features or aspects of the text …”, leading to inaccurate sentiment results when solely relying on textual data.

This limitation has driven the development of traditional multimodal approaches, where early fusion techniques within neural networks combine data from multiple sources (such as text, voice tone, and facial expressions). While these multimodal systems improve the understanding of emotions, they also introduce significant technical challenges, including handling noise, dealing with missing data, and interpreting the "black box" nature of neural networks. When one channel has a low confidence score or missing data, these systems can produce larger errors without fallback mechanisms to re-evaluate, which can affect overall accuracy. Additionally, neural networks are often opaque, making them difficult to interpret.

Our project proposes an alternative method: a voting mechanism applied after each channel has independently analyzed sentiment. This late fusion approach allows for greater interpretability and customizability, enabling researchers to handle edge cases such as conflicting emotional signals (e.g., facial expressions suggesting one emotion, while voice tone implies another). Using techniques like bagging or boosting, our voting system detects these conflicts and adjusts the final sentiment classification accordingly.

This system is not positioned as superior to state-of-the-art multimodal techniques. Instead, it serves as a complementary tool. In scenarios where multimodal systems work well, they should continue to be used. However, for edge cases and situations where there is ambiguity or conflict between signals, our voting mechanism provides a clearer, more transparent method of emotion detection. It allows users to combine both approaches: relying on multimodal systems for normal cases and utilizing our voting logic for more complex or nuanced cases.

Framework


Video of a person talking
Data separation
Video
Sound
Face sentiment analysis
Speech to text conversion
Voice tone sentiment analysis
Sentiment
Transcription
Sentiment
Speech sentiment analysis
Sentiment
Preprocessing and voting
Final predicted sentiment
Performance evaluation
Label
Performance metric

This section outlines the steps and the methodology used to extract and analyze sentiment from the video, combining the outputs from each data channel to derive a final sentiment classification. The flowchart above illustrates the workflow, which is explained in detail below.

1. Video of a person talking:

   The input to the system is a short, pre-recorded video file (approximately 30 seconds) containing both visual and auditory data of a single person speaking. The video must maintain consistent communication within a single context, avoiding misclassifications arising from abrupt changes in context or tone. These restrictions are set for the first iteration of our system to ensure that the core functionality is properly established. Future iterations may expand beyond these constraints, allowing for more complex analysis as the system evolves.

2. Data separation:

   The file is separated into two data types: Video which includes the visual representation of the person, particularly their facial expressions and movements, and sound which contains the auditory signals, including both the speech and tone of voice. These separated components are then fed into their respective processing units for sentiment analysis.

3. Face sentiment analysis:

   The video is processed to analyze facial expressions using machine learning models, commonly through face mesh or facial landmark techniques. This step detects micro expressions and overall facial movements to output a sentiment classification: positive, negative, or neutral.

4. Voice tone sentiment analysis:

   The tone of voice is analyzed to identify the sentiment based on vocal features such as pitch, volume, and intonation. This analysis helps in capturing sentiment that may not be explicit in the words themselves but is conveyed through tone. Again, output of this process is a sentiment classification.

5. Speech to text conversion:

The speech audio is processed through a speech-to-text algorithm, converting the spoken words into a textual transcription. This conversion allows for further semantic analysis, which is necessary for identifying the sentiment expressed in the speech content.

6. Speech sentiment analysis:

Once the speech is transcribed into text, it is analyzed to identify the sentiment based on linguistic patterns, word choice, and contextual clues in the speech. Again, output of this process is a sentiment classification.

7. Preprocessing and voting:

This process is the core of our project and will be the focus of development.

Before the voting mechanism, the system preprocesses each sentiment output from the three models to ensure that they are relevant and properly formatted for our algorithm. This includes tasks such as filtering, feature selection, and data cleaning.

The voting mechanism will employ a custom algorithm that weighs the contributions of each model, rather than simply using majority rule or mode. A key aspect of this mechanism will involve timestamp matching across channels, ensuring that the sentiment analysis from voice tone aligns with the corresponding transcription timestamps. This complex integration allows for a more nuanced evaluation of sentiment, as it takes into account when specific words were spoken in relation to vocal tone. Additionally, we will synchronize these analyses with facial expression timestamps as well. The output of this process is the final predicted sentiment.

8. Performance evaluation:

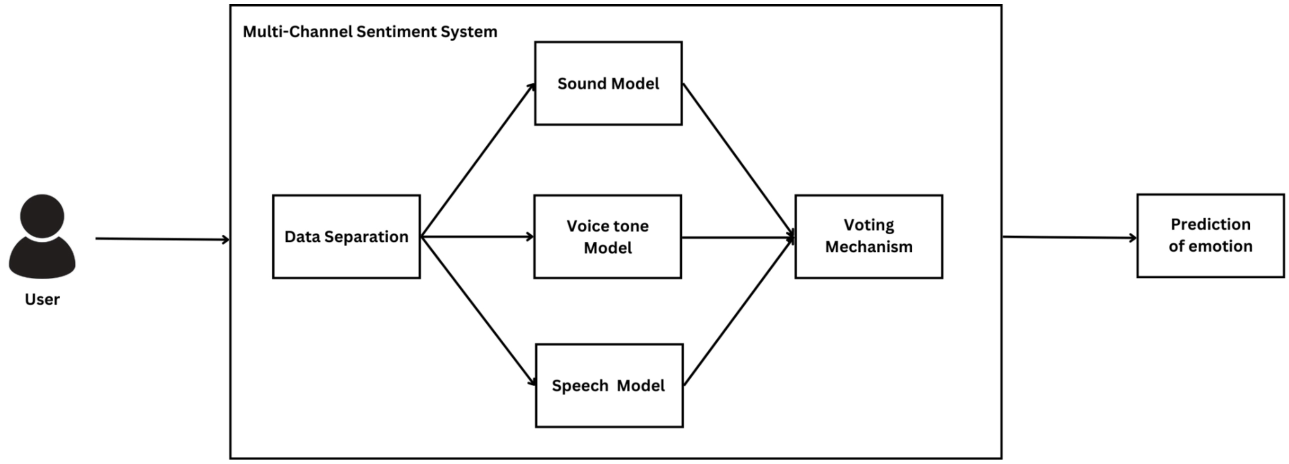
The system’s performance is evaluated using the F1-Score as the key metric, which balances precision and recall to give a comprehensive measure of the system's accuracy. To assess the accuracy, we compare the final predicted sentiment to the actual sentiment label, which is either manually determined by us reviewing the video or provided from the video source.

This process allows us to reflect on the model's performance and identify any necessary adjustments to the voting algorithm. For instance, if the model underperforms in certain scenarios, we may fine-tune the weights assigned to each model to improve the final sentiment prediction. Additionally, comparing the performance of our multi-channel system to single-modal systems (e.g., using only facial expressions or speech transcription) can help validate the benefits of the multi-channel approach that it indeed offers a more accurate understanding of emotional states.

Requirements Specification

Perspective

The interaction between the users and the web application is illustrated in the Figure below



From this figure, the system is operated by analyzing a user's live audio and video. The system extracts features from the voice tone, speech transcription, and facial expressions, and then uses different pre-trained machine-learning models to classify the user's overall emotion. After that, the system will weigh each result and vote for the final emotion prediction.

Requirements

The requirements of the system are listed in the following subsections.

Data Acquisition

DA1 The system should be able to capture and store voice recordings, facial images, and speech transcriptions from users.

DA2 The data acquisition process should be quick, ensuring that data is captured and processed promptly enabling timely emotion.

Data Preprocessing

DP1 The system should be preprocessed to extract relevant features such as pitch, intensity, and spectral characteristics.

DP2 Facial images should be preprocessed to detect and track facial landmarks, enabling the extraction of facial expressions.

DP3 Speech transcriptions should be cleaned and normalized to improve the quality of the input data for sentiment analysis.

Voting Mechanism

VM1 The system should employ a voting mechanism to combine the emotion classification results from the individual modalities.

VM2 The voting mechanism should assign weights to each modality based on their relative importance and reliability in the context of emotion detection.

VM3 The final emotion classification should be determined by our algorithm to weight with a score for each emotion conflict case.

Performance Evaluation

PE1 This system should be evaluated using appropriate metrics such as accuracy, precision, recall, and F1-score.

PE2 The performance of the system should be able to be compared to the existing single-channel emotion classification method like CNN architecture.

Preliminary Results

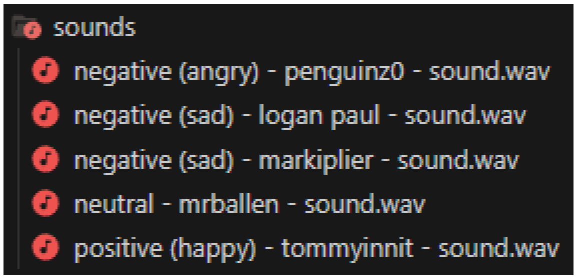
In this section, we present preliminary results from our initial testing of 2 of the sentiment analysis models: transcriptions and voice tone. The primary objective of this analysis was to gauge the effectiveness of using only a single data type in detecting sentiment from a video.

For our testing, we prepared a dataset comprising 5 audio samples from YouTube videos, each approximately 30 seconds long, and with background noise removed using UVR. These audio clips were specifically chosen to represent a range of emotional tones and contexts. The content audio clips content are as follows:

- Penguinz0 expressing anger through a rant about U.S. immigration services (https://youtu.be/8u-_Uh89R9w?si=LrikaIISKohnbPxW)
- Logan Paul apologizing for a uploading controversial a video of a dead body

(https://youtu.be/QwZT7T-TXT0?si=pJctHG1uG1dXlaTc)

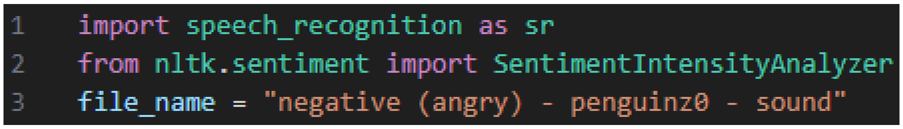
- Markiplier trying to stay positive while crying grieving a friend’s death (https://youtu.be/J_cxoZLPyR0?si=DG4iEZEDs5fC_Q9c)
- MrBallen narrating a love story, objectively (https://youtu.be/CBPYXcxyAOg?si=_pOui3tHBBZF3pRX)
- TommyInnit excitedly announcing his livestream plans (https://www.youtube.com/live/Op0X6a89He8?si=bcLARmFL6bQBTjMo)



```
sounds
  negative (angry) - penguinz0 - sound.wav
  negative (sad) - logan paul - sound.wav
  negative (sad) - markiplier - sound.wav
  neutral - mrballen - sound.wav
  positive (happy) - tommyinnit - sound.wav
```

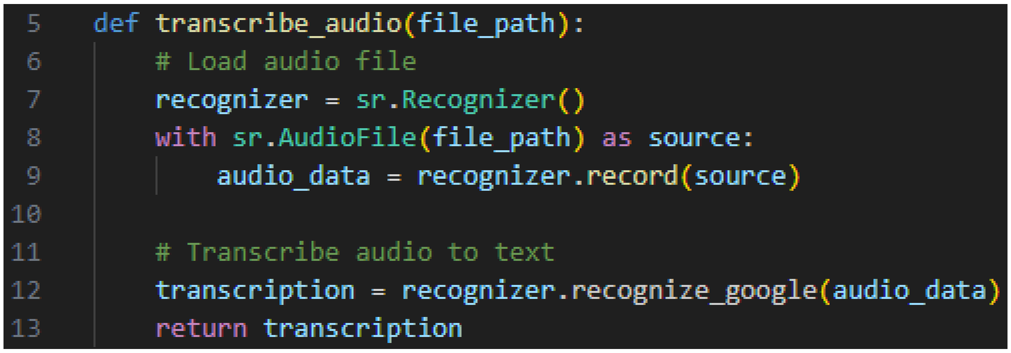
Transcription sentiment

For transcription sentiment analysis, we employed the sentiment_analyzer module from Python's Natural Language Toolkit (NLTK).



```python
import speech_recognition as sr
from nltk.sentiment import SentimentIntensityAnalyzer
file_name = "negative (angry) - penguinz0 - sound"
```

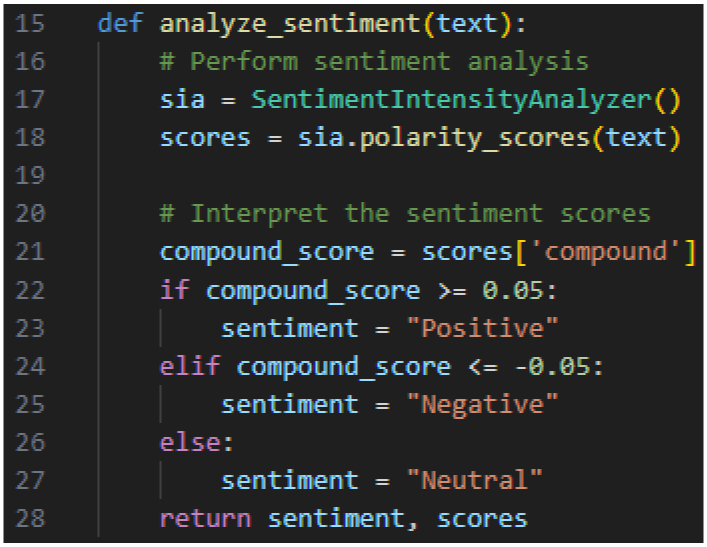
To convert the speech audio into text, we utilized the Google Speech Recognition API, accessed through the SpeechRecognition library in Python. This API prepares the input necessary for sentiment analysis.



```python
def transcribe_audio(file_path):
    # Load audio file
    recognizer = sr.Recognizer()
    with sr.AudioFile(file_path) as source:
        audio_data = recognizer.record(source)

    # Transcribe audio to text
    transcription = recognizer.recognize_google(audio_data)
    return transcription
```

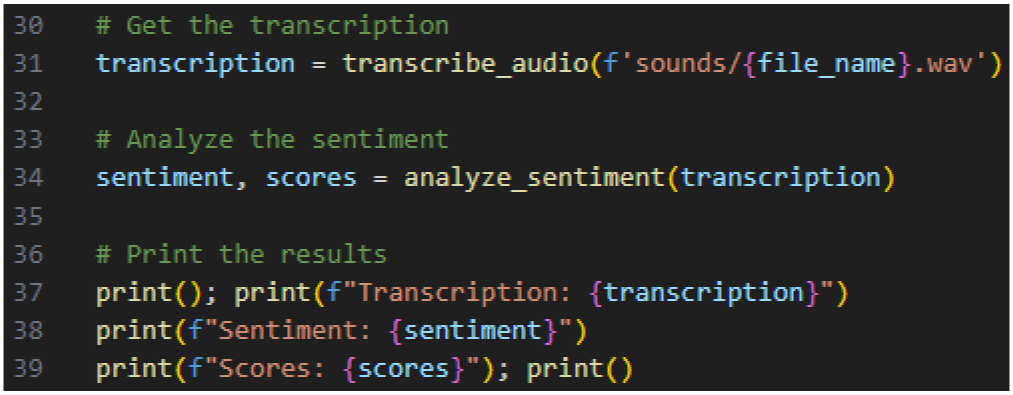
Upon successful transcription of the audio samples, we analyzed the sentiment using the NLTK's SentimentIntensityAnalyzer. This analyzer employs a lexicon-based approach to assess sentiment, providing a score that reflects the positive, negative, and neutral sentiment contained within the transcription. Each audio sample's sentiment scores were recorded for further evaluation. With the compound score value representing an overall sentiment.



```python
def analyze_sentiment(text):
    # Perform sentiment analysis
    sia = SentimentIntensityAnalyzer()
    scores = sia.polarity_scores(text)

    # Interpret the sentiment scores
    compound_score = scores['compound']
    if compound_score >= 0.05:
        sentiment = "Positive"
    elif compound_score <= -0.05:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"
    return sentiment, scores
```

Next, we run the following code for each audio sample.



```python
# Get the transcription
transcription = transcribe_audio(f'sounds/{file_name}.wav')

# Analyze the sentiment
sentiment, scores = analyze_sentiment(transcription)

# Print the results
print(); print(f"Transcription: {transcription}")
print(f"Sentiment: {sentiment}")
print(f"Scores: {scores}"); print()
```

<u>negative (angry) - penguinz0</u>

Transcription: it's not possible because there is no reason we would keep being declined they're not even giving us reasons that make any sense at all not to us not to the lawyers not anybody nobody and no other Esports team has had this problem that has international players got in lickety split mean we can't we literally can't

Sentiment: Negative

Scores: {'neg': 0.15, 'neu': 0.85, 'pos': 0.0, 'compound': -0.8206}

negative (sad) - logan paul

Transcription: they were unfiltered none of us knew how to react or how to feel I should have never posted the video I should have put the cameras down and stopped recording what we were going through there's a lot of things I should have done differently but I didn't

Sentiment: Negative

Scores: {'neg': 0.033, 'neu': 0.967, 'pos': 0.0, 'compound': -0.1154}

negative (sad) - markiplier

Transcription: in in small ways that you probably didn't know Daniel was part of what I did a big part of what I did and Daniel he did a lot of things that were often unseen and I just hope that he knew how much I respected him as a friend

Sentiment: Positive

Scores: {'neg': 0.0, 'neu': 0.813, 'pos': 0.187, 'compound': 0.8481}

neutral (calm) - mrballen

Transcription: Mitch was in his final semester at a college in Louisiana when he met another senior a wonderful young lady named Kayla from the instant he saw her he knew he was in love she played hard to get at first but after several months he won her over after graduation the pair stayed in Louisiana and moved in together while they got their careers off the ground two years later the pair got married and almost immediately Kayla got pregnant at the time Mitch's career was really starting to take off which allowed Kayla to stay at home and take some time off

Sentiment: Positive

Scores: {'neg': 0.011, 'neu': 0.883, 'pos': 0.106, 'compound': 0.8885}

positive (happy) - tommyinnit

Transcription: but today I've been in a photo shoot all day but I'm very knackered you know but I wore the red and white top and I got on the stream and I said fuck it dude I'm actually like really giddy we're going to hop on Hypixel I have some plans for us that we're going to do you guys some things that I say we need to do

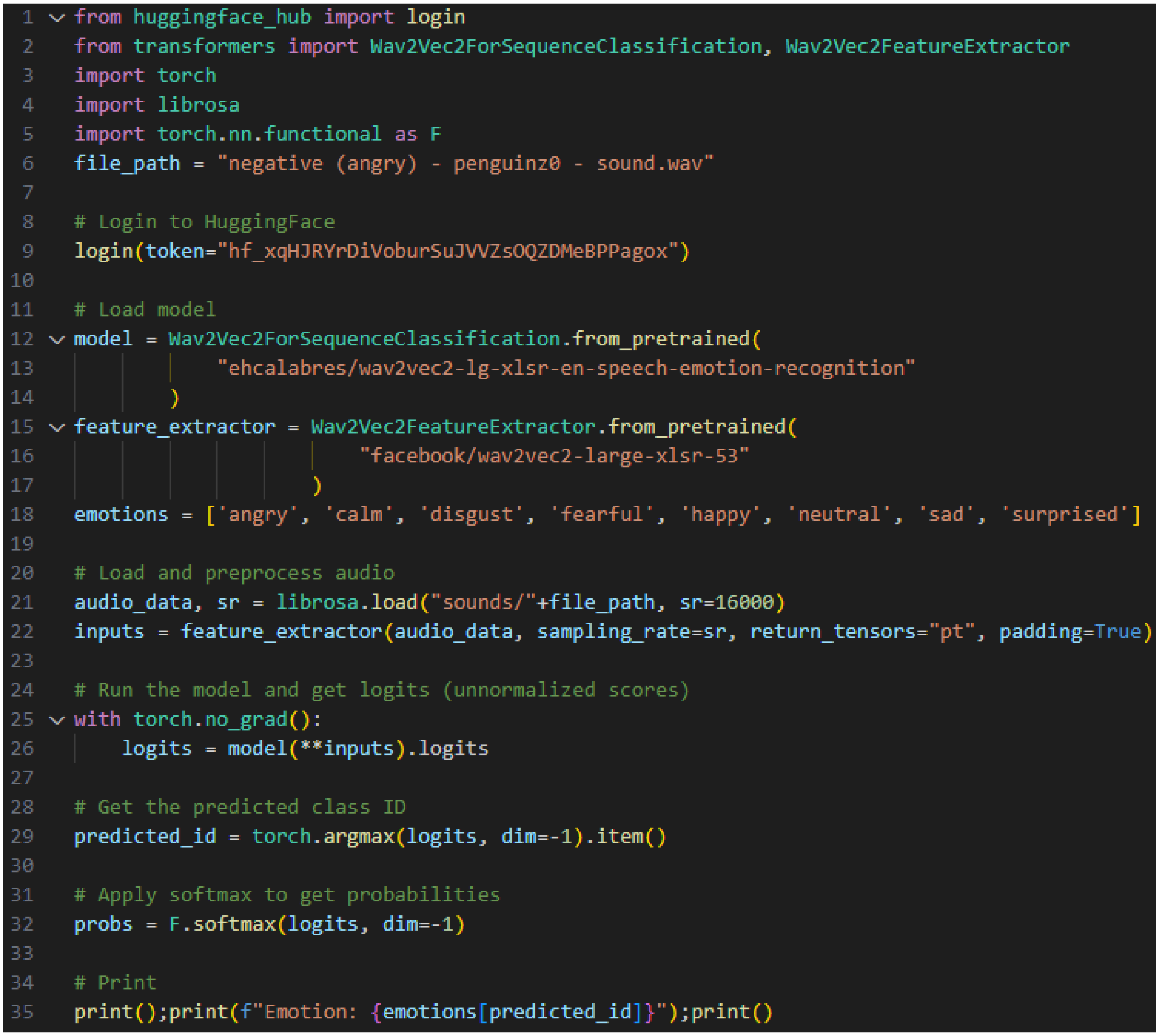
Sentiment: Positive

Scores: {'neg': 0.045, 'neu': 0.875, 'pos': 0.08, 'compound': 0.3291}

These observations highlight the limitations of using only textual data for sentiment analysis in speech. It struggled with nuanced emotions and complex contexts. Markiplier's somber speech on loss was misclassified as positive, likely due to words like "respect" and "hope" in the transcription even though he’s crying. Similarly, MrBallen's neutral storytelling was interpreted as strongly positive, misled by terms like "wonderful" and "love." Subtle negative emotions proved challenging to identify accurately. Logan Paul's apology, while correctly labeled negative, received a weak negative score, suggesting difficulty in detecting remorse or regret when not expressed through overtly negative language. The analysis of Tommyinnit's excited announcement, though correctly classified as positive, failed to fully capture the enthusiasm evident in the audio. This highlights the system's inability to account for vocal tone and energy levels.

Voice tone sentiment

For voice tone sentiment analysis, we employed wav2vec2-lg-xlsr-en-speech-emotion-recognition, wav2vec2-large-xlsr-53 models from huggingface. The dataset used to fine-tune the original pre-trained model is the RAVDESS dataset. This dataset provides 1440 samples of recordings from actors performing on 8 different emotions in English, which are: ['angry', 'calm', 'disgust', 'fearful', 'happy', 'neutral', 'sad', 'surprised']. Unlike transcription-based sentiment analysis, which relies on lexical cues and classifies sentiment primarily into positive, negative, and neutral categories, voice tone sentiment analysis identifies a broader range of emotions directly from audio, regardless of spoken content. This allows it to capture the nuances of speech such as pitch, intonation, and energy levels.



```python
from huggingface_hub import login
from transformers import Wav2Vec2ForSequenceClassification, Wav2Vec2FeatureExtractor
import torch
import librosa
import torch.nn.functional as F
file_path = "negative (angry) - penguinz0 - sound.wav"

# Login to HuggingFace
login(token="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")

# Load model
model = Wav2Vec2ForSequenceClassification.from_pretrained(
            "ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition"
        )
feature_extractor = Wav2Vec2FeatureExtractor.from_pretrained(
                        "facebook/wav2vec2-large-xlsr-53"
                    )
emotions = ['angry', 'calm', 'disgust', 'fearful', 'happy', 'neutral', 'sad', 'surprised']

# Load and preprocess audio
audio_data, sr = librosa.load("sounds/"+file_path, sr=16000)
inputs = feature_extractor(audio_data, sampling_rate=sr, return_tensors="pt", padding=True)

# Run the model and get logits (unnormalized scores)
with torch.no_grad():
    logits = model(**inputs).logits

# Get the predicted class ID
predicted_id = torch.argmax(logits, dim=-1).item()

# Apply softmax to get probabilities
probs = F.softmax(logits, dim=-1)

# Print
print();print(f"Emotion: {emotions[predicted_id]}");print()
```

The models were applied to the same set of audio samples used in transcription sentiment analysis to evaluate the emotional tone conveyed in each speech. Below are the results:

negative (angry) - penguinz0

Detected Emotion Audio Tone: angry

negative (sad) - logan paul

Detected Emotion Audio Tone: disgust

negative (sad) - markiplier

Detected Emotion Audio Tone: happy

neutral (calm) - mrballen

Detected Emotion Audio Tone: surprised

positive (happy) - tommyinnit

Detected Emotion Audio Tone: happy

These observations highlight the limitations of using only voice tone data for sentiment analysis. While the model accurately captured strong emotions like Penguinz0's anger and TommyInnit's happiness, it struggled with more nuanced or layered emotions. Logan Paul's apology was misclassified as disgust rather than sadness, reflecting the model's difficulty in distinguishing remorse from other negative emotions. Markiplier's shaking voice, despite expressing grief, was misclassified as happy somehow, failing to detect the underlying sadness. Similarly, MrBallen's neutral storytelling was incorrectly labeled as surprised, indicating over-interpretation of subtle tonal shifts.

When comparing voice tone and transcription sentiment analysis, both modalities show significant limitations in capturing the full emotional spectrum of speech. While transcription-based analysis misinterpreted emotionally charged words, voice tone struggled with subtle or layered emotions like regret, sadness, or calmness mixed with sorrow. Relying on either modality alone proves insufficient for accurate sentiment detection, especially in real-world contexts where emotions are complex and layered.

Facial data from video can help address these gaps. Facial expressions provide valuable non-verbal cues, such as a furrowed brow or teary eyes, that can signal sadness, anger, or disgust more accurately than voice tone or transcription alone. For instance, in Markiplier's case, the sadness in his teary eyes and facial expressions would clarify the emotional tone that neither voice nor words could fully convey.

Bibliography

1. Poria, S., Majumder, N., Mihalcea, R., & Hovy, E. (n.d.). Emotion recognition in conversation: Research challenges, datasets, and recent advances. Retrieved from https://paperswithcode.com/paper/emotion-recognition-in-conversation-research
2. Seunghyun Yoon, Seokhyun Byun, & Kyomin Jung. Multimodal Speech Emotion Recognition Using Audio and Text [Papers With Code]. Retrieved from https://paperswithcode.com/paper/multimodal-speech-emotion-recognition-using
3. Tzirakis, P., Trigeorgis, G., Nicolaou, M. A., Schuller, B., & Zafeiriou, S. (n.d.). End-to-end multimodal emotion recognition using deep neural networks. Retrieved from https://paperswithcode.com/paper/end-to-end-multimodal-emotion-recognition
4. Sangineto, A., Liu, H., & Xu, L. (2020, September). Nonparallel emotional speech conversion using VAE-GAN. In INTERSPEECH 2020 (pp. 1043-1047). International Speech Communication Association. Retrieved from https://paperswithcode.com/paper/nonparallel-emotional-speech-conversion
5. Hazarika, D., Poria, S., Zimmermann, R., & Mihalcea, R. (2019). Conversational transfer learning for emotion recognition. Retrieved from https://paperswithcode.com/paper/emotion-recognition-in-conversations-with