Sentiment Analysis

Prototype

Satyam Chand Deopa

1. **Objective:**

The objective of this app is to analyze and deduct the sentiment contained within many forms of data like text, voice, images, and video. Knowing these factors, especially for the customers, clients can help a company infer not only what kind of feelings do their consumers/clients have of their products, but also how it is perceived by the general public.

This project is made as a submission to TCS for the Inframind Season-4.

1. **Modules:**

**Text:** The text-based sentiment analysis is done using a lemmatized dictionary. The sentences (or paragraphs) are tokenized suing the nltk tokenizer and the tokenized dictionary is then passed to the model which predicts the sentiment. The model trained under Sentiment140 dataset which contained over 1.6 million labeled tweet examples.

**Speech:** The speech-based sentiment analysis is done by adding some functionalities over the text-based sentiment analysis. The speech recognition is done (currently available through .mp3 files) by reading a file and implementing speech to text translation using Speech Recognition library. This now text formatted data is then passed over to the text-based recognizer to get the outcome of the sentiment.

**Image:**  The Image based sentiment recognition is done by using Facial Expression Recognition (fer) library. The library by default uses Open-CV’s Haar Cascade Classifier to detect the faces. These faces are then passed into the Convolutional Neural Network for sentiment detection.

**Video:** A video analysis is carried out in a similar fashion to the image sentiment analysis. However, in this case we first extract the frames from the source whether it is a video file or is accessed through webcam. These frames are further streamed to analyze and get the sentiment of faces inside them.

**Twitter:** We can analyze tweets in our app using the Twitter’s API v1.1(Standard). We can skim a twitter timeline to get the statues of a specified user or even get related content. Currently, we can get the timeline of a user. This is implemented using a cursor and for each iteration, we can extract the required info and then analyze the sentiment of the text in the status.

**To keep an account**

To keep an idea of the sentiments analyzed by us, a couple of counters have been implemented as dictionaries to keep track of them and are visible in the results page. These are:

**Global sentiment**: This gives a general overview of the sentiments occurring in the app as a whole. It is based on two levels: Positive and Negative.

**Visual media sentiment:** This dictionary gives an estimate of occurrence of sentiment contained in image or video format. Since the analysis is done using FER, the dictionary in split in 7 factors. These are angry, disgust, fear, happy, neutral, sad and surprise.

The counter increment is done in in following manner:

Image: 100 counts

Video: 1 count per frame.

Also, the conversion for global emotion counter is done in the manner:

1 count of Global sentiment = 100 counts of visual media sentiment

1. **Media formats:**

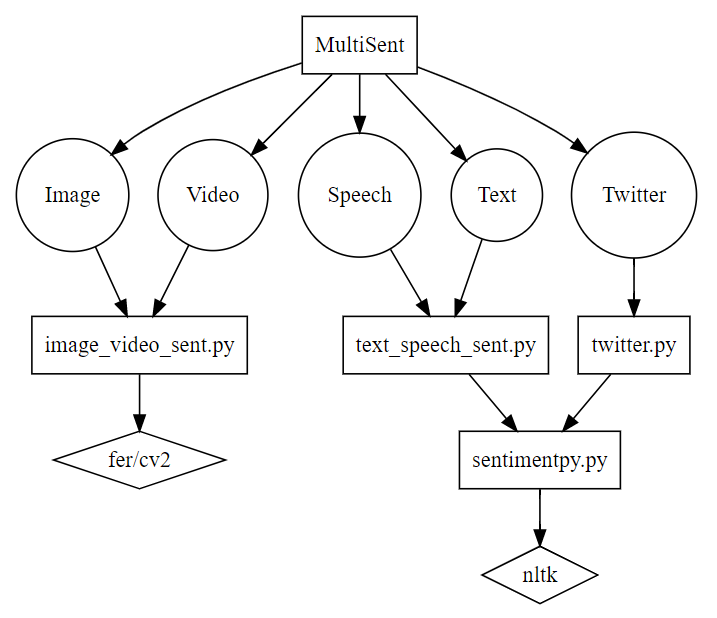
Text: txt

Speech: wav

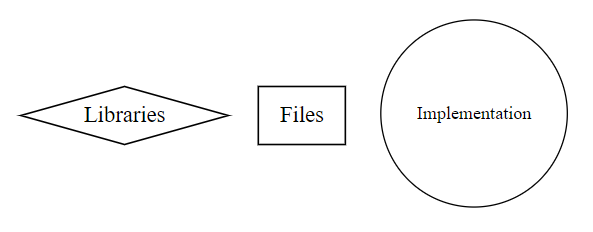
Image: jpg, jpeg, png

Video: mkv

1. **File Dependencies:**



Legend:



**Links:**

Video URL: <https://youtu.be/z6IQEhlZk0s>

GitHub Project Link: <https://github.com/SatyamChand/MultiSent>

LinkedIn: <https://www.linkedin.com/in/satyamchand/>

E-mail: [satyamdeopa@gmail.com](mailto:satyamdeopa@gmail.com)

Thank You!