Audio Sentiment Analysis Using Deep Learning

A thesis submitted in partial fulfilment of the requirement for the degree of
Master of Technology
in
Computer Technology

Submitted by
Anupam Barat
Registration No.: 154187 of 2020-2021,
Examination Roll No.: M3TCT22006
Session: 2020-2023

Under the Supervision of
Prof. Debotosh Bhattacharjee
Department of Computer Science and Engineering
Jadavpur University,
188, Raja S.C. Mallick Rd,
Kolkata - 700032,
West Bengal, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that this is a bonafide record of the project entitled **"Audio Sentiment Analysis Using Deep Learning"**, submitted by Anupam Barat (University Registration No.: 154187 of 2020-2021, Examination Roll No.: M3TCT22006), is hereby approved of a creditable study of a technological subject carried out under my supervision and presented in a manner satisfactory to warrant its acceptance for partial fulfilment of the requirements of the degree of Master of Technology in Computer Technology. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other university or institute.

Supervisor

. . . . . .

Prof. Debotosh Bhattacharjee
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

Countersigned

. . . . . .

Prof. Nandini Mukherjee
Head, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

. . . . . .

Prof. Ardhendu Ghoshal
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata-32, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval[1]

This is to certify that the thesis entitled "**Audio Sentiment Analysis Using Deep Learning**", is a bonafide record of work carried out by Anupam Barat in partial fulfilment of the requirements of the degree of Master of Technology in Computer Technology in the Department of Computer Science and Engineering, Jadavpur University during the period of December 2020 to June 2023. It is understood that by this approval, the undersigned do not necessarily endorse any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

. . . . . .
Signature of Examiner 1
Date:

. . . . . .
Signature of Examiner 2
Date:

_____
Only in case the thesis is approved[1]

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "**Audio Sentiment Analysis Using Deep Learning**" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Technology in Computer Technology.

All information has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Anupam Barat

Registration No.: 154187 of 2020-2021

Examination Roll No.: M3TCT22006

Thesis Title: Audio Sentiment Analysis Using Deep Learning

. . . . . .
Signature with date

ACKNOWLEDGEMENT

I would like to acknowledge and give my warmest thanks to my supervisor **Prof. Debotosh Bhattacharjee,** who made this work possible. His guidance and advice carried me through all the stages of writing my project.

Most importantly, none of this could have happened without the love and support of my family. To **my father Mr. Dhiraj Prasad Barat, my mother Mrs. Sampa Barat, my sister Ankhi Barat** and **my brother Mr. Souvik Biswas** – it would be an understatement to say that their unconditional love and encouragement has always helped me in my need. With their forbearance and whole-hearted support, this thesis would not have been able to see the light of the day.

Finally, I would like to thank God for letting me through all the difficulties. I have experienced your guidance day by day. I will keep on trusting you for my future.

. . . . . .

Anupam Barat
Examination Roll No.: M3TCT22006
Dept. of Computer Science & Engineering
Jadavpur University
Kolkata, India

Abstract

The challenging issue of sentiment analysis in natural audio sources is the subject of the research that is being suggested, which focuses particularly on speaker-discriminated speech transcripts. The objective is to identify the emotional states that each speaker in a conversation exhibits. This study acknowledges that current methods for sentiment extraction frequently rely on text-based sentiment classifiers, which might not be able to capture the subtleties and emotional cues inherent in raw audio.

The study investigates several strategies for speaker discrimination and sentiment analysis to meet this goal. Identification and distinction of various speakers in an audio discussion are accomplished by speaker discrimination. The importance of this phase lies in the fact that it enables the sentiment analysis to be conducted on certain speakers rather than the entire conversation.

The research most likely entails using labelled audio datasets that have been manually annotated with speaker names and accompanying sentiment labels to train deep learning models. The models may use a variety of neural network architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), that are appropriate for processing audio data. Further methods, including feature extraction, audio modelling, and language modelling may be investigated to improve the sentiment analysis process.

The long-term goal of this project is to create effective algorithms capable of reliably analysing sentiment on speaker-discriminated voice transcripts. These algorithms would make it possible to automatically identify different speakers' emotions, adding to the growing field of audio sentiment analysis.

In this proposed research, we perform sentiment analysis on speaker-discriminated speech transcripts to detect the emotions of the individual speakers involved in the conversation. We analyzed different techniques to perform speaker discrimination and sentiment analysis to find efficient algorithms to perform this task.

Contents

List of Figures and Tables

Chapter-1 Introduction

Indeed, sentiment analysis is essential for determining how individuals feel and act in various situations. A system that can identify both the speakers involved and the spoken content must be put in place if machines can understand human thinking through talks. Before performing sentiment analysis on the retrieved data, this includes creating a speaker and speech recognition system.

Understanding human emotions can be quite useful in a variety of situations. For instance, computers can recognise and react to user wants using non-verbal communication, such as emotions. By recognizing human emotions, machines can improve user experience by personalising settings and answers based on preferences.

The scientific community has already looked into ways to convert audio files like songs, discussions, news stories, and political disputes into text for study. Investigations have also been done to listen to audio recordings of multi-speaker meetings and phone calls to customer care. However, it cannot be easy to analyse audio recordings with several speakers.

The authors of this study suggest creating TensorFlow Keras models to categorise sentiment in audio recordings as positive, negative, or neutral. The study most likely compares various models, their setups, and the output outcomes of testing the models.

The fact that emotional information is frequently constrained to a single modality is a major obstacle in the recognition of emotions. Relying entirely on a single modality makes it impossible to appropriately assess emotional states, given the growing accessibility of audio data on social media. For instance, textual sentiment analysis can only evaluate emotions that are communicated by words, phrases, sentences and the relationships among them. This method frequently falls short of capturing the whole emotional content. In regular interactions, voice is routinely added to text, though social media sites have a close relationship with text and speech. Voice can communicate a multitude of information because it is the most direct way of communication. Researchers have achieved substantial advancements in identifying spoken emotions by examining the connection between text and voice.

The suggested research involves using modal fusion, which combines the results of voice emotion analysis and text-based sentiment analysis, to enhance the performance of social media emotional detection systems. The final emotional state can be ascertained by considering both speech and text, resulting in a more thorough comprehension of emotions.

Deep learning models are probably trained on labelled audio datasets manually annotated with speaker identifiers and accompanying sentiment labels to carry out this research. The models may use a variety of neural network architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), that are appropriate for processing audio data. Methods including feature extraction, audio modelling, and language modelling may be researched to further improve the sentiment analysis procedure.

Chapter-2 Related Work and Background Overview

Sentiment analysis is essential for gathering and analysing people's opinions, ideas, and first impressions about a variety of themes, goods, and services. For businesses, governments, and individuals, the insights gained from sentiment analysis can be useful in enabling them to make educated decisions and take the proper measures depending on public opinion.

However, given the wide range of media sources, including text, audio, photos, and videos, sentiment analysis does confront certain difficulties. The challenges of effectively gathering and analysing sentiment data from each of these media forms are distinct.

The method that has been the subject of the most research and application is text-based sentiment analysis. In order to ascertain the sentiment polarity (positive, negative, or neutral) communicated in the text, it entails analysing textual data from several sources, including social media postings, customer reviews, news articles, and more. But sarcasm, irony, context-dependent feelings, and cultural quirks pose problems for proper interpretation in text-based sentiment analysis.

Sentiment analysis must take into account audio, image, and video content in addition to text. To extract and comprehend the sentiments indicated in the speech, techniques like speech recognition and emotion recognition must be used when analysing sentiment in audio recordings. Analysing visual content to extract emotions or sentiment-related features from photos or video frames is similar to the image and video sentiment analysis.

Each of these media kinds poses a unique set of difficulties. For instance, problems with background noise, speaker variability, and speech recognition errors might affect the precision of sentiment extraction in audio sentiment analysis. Similar difficulties must be overcome in facial expression, body language, and contextual cue recognition in picture and video sentiment analysis.

The current research focus is overcoming these difficulties and correctly assessing attitudes in various media sources. To improve sentiment analysis across many media forms, researchers and practitioners are constantly developing and improving machine learning methods and deep learning approaches. By overcoming these obstacles, sentiment analysis can offer insightful sentiment and public opinion data, assisting various decision-making processes.

2.1 Sentiment Analysis for various media

Due to the immense use of the internet and various social media every day, a massive quantity of data is produced and disseminated through various channels and sentiment analysis should consider all these data for opinion mining and other purposes essential for an effective, productive system.

2.1.1 Sentiment analysis from text data

Sentiment analysis from text data involves extracting and analyzing the sentiment or emotions expressed in the textual content. This technique aims to understand the subjective information and sentiment polarity (positive, negative, or neutral) conveyed in the text.

The process of sentiment analysis from text data typically involves the following steps:

1. Data Collection: Gathering text data from various sources such as social media platforms, customer reviews, surveys, news articles, or any other text-based content relevant to the analysis.

2. Text Preprocessing: Cleaning and preparing the text data for analysis by removing irrelevant information, punctuation, special characters, and stopwords (commonly used words with little sentiment value). This step may involve tokenization (splitting text into individual words or tokens) and stemming or lemmatization (reducing words to their base form).

3. Feature Extraction: Transforming the text data into numerical features that can be used for sentiment analysis. This can be done using techniques like bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings (e.g., Word2Vec, GloVe), or more advanced approaches like BERT (Bidirectional Encoder Representations from Transformers) for contextual word representations.

4. Sentiment Classification: Applying a sentiment classification algorithm to assign sentiment labels to the text data. This can be done using machine learning techniques such as Naive Bayes, Support Vector Machines (SVM), Random Forests, or more sophisticated deep learning models like recurrent neural networks (RNNs) or transformer-based models.

5. Model Evaluation: Assessing the performance of the sentiment analysis model using appropriate evaluation metrics like accuracy, precision, recall, F1-score, or area under the ROC curve (AUC-ROC).

6. Post-processing and Visualization: Analysing the results, interpreting the sentiment classifications, and visualizing the sentiment distribution or trends using techniques like word clouds, bar charts, or sentiment heatmaps.

Sentiment analysis from text data finds applications in a wide range of domains, including customer feedback analysis, brand monitoring, social media sentiment analysis, market research, political analysis, and more. It enables organizations to understand the public sentiment, customer opinions, and emerging trends, helping them make data-driven decisions and tailor their strategies accordingly.

2.1.2 Sentiment analysis from speech

Sentiment analysis from speech involves extracting and analyzing the sentiment or emotions expressed in spoken language or audio recordings. It aims to understand the subjective information and sentiment polarity conveyed through speech.

The process of sentiment analysis from speech typically involves the following steps:

1. Speech Recognition: Converting the spoken language or audio recordings into textual transcripts. Automatic Speech Recognition (ASR) systems are used to transcribe the speech into text, which serves as the input for subsequent sentiment analysis.

2. Text Preprocessing: Cleaning and preparing the transcribed text data for sentiment analysis. This may involve removing irrelevant information, punctuation, special characters, and stopwords. Tokenization and stemming or lemmatization techniques may also be applied to preprocess the text.

3. Feature Extraction: Transforming the preprocessed text data into numerical features that can be used for sentiment analysis. This can be done using techniques like bag-of-words, TF-IDF, or word embeddings.

4. Sentiment Classification: Applying a sentiment classification algorithm to assign sentiment labels to the text data obtained from speech. This can be done using machine learning techniques such as Naive Bayes, SVM, Random Forests, or deep learning models like RNNs or transformer-based models.

5. Model Evaluation: Assessing the performance of the sentiment analysis model using evaluation metrics such as accuracy, precision, recall, F1-score, or AUC-ROC.

6. Post-processing and Visualization: Analyzing the sentiment classifications, interpreting the sentiment results, and visualizing the sentiment distribution or trends. This may involve generating visualizations such as sentiment heatmaps or sentiment timelines.

Sentiment analysis from speech has applications in various domains, including customer service analysis, call center monitoring, voice-based sentiment analysis in social media, market research, and more. It enables organizations to gain insights into customer sentiments and emotions expressed through spoken language, helping them improve customer experiences, identify trends, and make informed decisions based on sentiment analysis results.

2.1.3 Visual sentiment analysis

Visual sentiment analysis is the process of analyzing and determining the sentiment or emotions expressed in visual content, such as images or videos. It involves using computer vision techniques to extract features from visual data and applying machine learning algorithms to classify the sentiment conveyed by the visual content.

The main steps in visual sentiment analysis include the following:

1. Data Collection: Gathering a dataset of images or videos that are labeled with sentiment or emotion categories.

2. Feature Extraction: Extracting visual features from images or videos. This can involve techniques such as extracting color histograms and texture features or using deep learning models like convolutional neural networks (CNNs) to extract high-level visual representations.

3. Sentiment Classification: Applying machine learning or deep learning algorithms to classify the visual content into sentiment or emotion categories. This can be done using approaches such as support vector machines (SVM), random forests, or deep neural networks.

4. Model Evaluation: Assessing the performance of the sentiment analysis model using appropriate evaluation metrics like accuracy, precision, recall, F1-score, or AUC-ROC.

5. Post-processing and Visualization: Analyzing the sentiment analysis results, interpreting the sentiment labels assigned to the visual content, and visualizing the sentiment distribution or trends. This can involve generating visualizations such as sentiment heatmaps, sentiment-based image/video retrieval, or sentiment timelines.

Visual sentiment analysis finds applications in various fields, including social media analysis, brand monitoring, advertisement analysis, and multimedia content understanding. It enables the automatic analysis of sentiment expressed through visual content, allowing organizations to gain insights into public sentiment, user preferences, and emotional responses to visual stimuli.

2.1.4 Multimodal Sentiment Analysis

Multimodal sentiment analysis refers to the analysis of sentiment or emotions by combining and integrating information from multiple modalities, such as text, audio, and visual data. It aims to capture a more comprehensive understanding of sentiment by leveraging the rich and complementary information available in different modalities.

The process of multimodal sentiment analysis typically involves the following steps:

1. Data Collection: Gathering a dataset that includes multiple modalities, such as text, audio, and visual data. The dataset should be labeled with sentiment or emotion categories.

2. Modality-specific Preprocessing: Preprocessing each modality individually to prepare the data for analysis. This can involve text preprocessing techniques for text data, audio feature extraction for audio data, and visual feature extraction for visual data.

3. Fusion of Modalities: Integrating the information from different modalities to capture the interactions and dependencies between them. Fusion techniques can range from simple concatenation or weighted combination of features to more sophisticated methods like late fusion, early fusion, or multi-modal deep learning models.

4. Sentiment Classification: Applying machine learning or deep learning algorithms to classify the multimodal data into sentiment or emotion categories. This can involve training models that take into account the fused representations from multiple modalities.

5. Model Evaluation: Assessing the performance of the multimodal sentiment analysis model using appropriate evaluation metrics. These metrics can measure the sentiment predictions' accuracy, precision, recall, F1-score, or AUC-ROC.

Post-processing and Visualization: Analyzing the sentiment analysis results, interpreting the multimodal sentiment predictions, and visualizing the sentiment distribution or trends across different modalities.

Multimodal sentiment analysis has various applications, including social media analysis, movie or product reviews, human-computer interaction, and affective computing. Incorporating information from multiple modalities enhances the accuracy and robustness of sentiment analysis by capturing nuanced sentiment expressions and overcoming the limitations of analyzing a single modality alone.

2.2 Sentiment Analysis Techniques

Sentiment analysis techniques can be broadly categorized into the following types:

2.2.1. Lexicon-Based Techniques

These techniques rely on sentiment lexicons or dictionaries containing pre-defined sentiment scores or word labels. The sentiment of a given text is determined by aggregating the sentiment scores of the individual words in the text. Lexicon-based techniques are relatively simple and efficient but may lack contextual understanding.

2.2.2. Machine Learning-Based Techniques

These techniques involve training a machine learning model on labeled sentiment data to predict the sentiment of the unseen text. Some commonly used machine learning algorithms for sentiment analysis include Naive Bayes, Support Vector Machines (SVM), Random Forest, and Logistic Regression. These techniques require labeled training data and can capture more complex relationships and contextual information.

2.2.3. Deep Learning-Based

Techniques Deep learning techniques, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformer models, have succeeded in sentiment analysis. These models can capture text data's sequential dependencies, semantic relationships, and contextual information. They require large amounts of labeled training data and can be computationally expensive but achieve high accuracy.

2.2.4. Rule-Based Techniques

Rule-based techniques use a set of predefined rules or patterns to identify sentiment in text. These rules can be based on linguistic rules, syntactic patterns, or regular expressions. Rule-based techniques can be useful for specific domains or languages but may lack generalizability.

2.2.5. Hybrid Approaches

Hybrid approaches combine the abovementioned techniques to improve the accuracy of sentiment analysis. For example, a hybrid approach may utilize lexicon-based techniques for initial sentiment classification and refine the results using machine learning or deep learning models.

It's important to note that each technique has its strengths and weaknesses, and the choice of technique depends on factors such as available resources, data availability, domain-specific requirements, and desired performance. Researchers and practitioners often experiment with different techniques to find the most suitable approach for their sentiment analysis task.

2.3 Audio Sentiment Analysis Techniques

Audio sentiment analysis techniques aim to detect and analyze the sentiment or emotion expressed in spoken language or audio recordings. Here are some common techniques used in audio sentiment analysis:

2.3.1. Acoustic Feature Extraction

Audio signals are transformed into acoustic features that capture various sound characteristics. These features may include pitch, energy, tempo, spectral features, and prosodic features like intonation and rhythm. Acoustic features provide information about the speaker's voice, tone, and expression, which can indicate the underlying sentiment.

2.3.2. Speech Recognition

Automatic Speech Recognition (ASR) systems convert spoken language into text. In audio sentiment analysis, ASR can transcribe the spoken content, which can then be processed using text-based sentiment analysis techniques. ASR-based sentiment analysis requires a separate speech recognition module before sentiment analysis can be applied.

2.3.3. Lexicon-Based Approaches

Lexicon-based techniques can be extended to audio sentiment analysis by mapping acoustic features to sentiment lexicons. Acoustic features such as pitch, intensity, and voice quality can be associated with sentiment categories to estimate sentiment scores. Lexicon-based

approaches in audio sentiment analysis often combine acoustic features with linguistic features extracted from the speech transcript.

2.3.4. Machine Learning and Deep Learning

Similar to text-based sentiment analysis, machine learning, and deep learning algorithms can be applied to audio sentiment analysis. These models can directly process acoustic features or use a combination of acoustic and linguistic features extracted from speech transcripts. Popular machine learning models such as Support Vector Machines (SVM), Random Forests, or deep learning models like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can be trained on labeled audio data to predict sentiment labels.

2.3.5. Multimodal Fusion

Audio sentiment analysis can benefit from multimodal fusion, which combines information from multiple modalities such as audio, video, and text. More robust sentiment predictions can be made by integrating acoustic features with visual cues from facial expressions or text-based sentiment analysis.

2.3.6. Transfer Learning

Transfer learning techniques can be utilized in audio sentiment analysis by leveraging pre-trained models on large speech-related datasets. By fine-tuning or adapting these models to the specific sentiment analysis task, they can capture higher-level representations of audio sentiment and improve performance.

It is worth noting that audio sentiment analysis is a challenging task due to the complexity of acoustic data and the inherent variability in speech. The choice of technique depends on the application's specific requirements, the availability of labeled audio data, and the desired level of accuracy and interpretability. Researchers and practitioners continue to explore and develop new techniques to enhance the performance of audio sentiment analysis systems.

Chapter-3 Proposed Technique

We chose the deep learning technique for audio sentiment analysis due to several reasons:

1. Ability to capture complex patterns: Deep learning models, such as CNN, GRU, and LSTM, have proven highly effective in capturing intricate patterns and dependencies in audio data. These models can automatically learn and extract relevant features from raw audio signals, enabling better representation and understanding of the underlying sentiment.

2. Handling sequential and temporal data: Audio data is inherently sequential and temporal, consisting of a sequence of time-dependent samples. Traditional machine-learning techniques may struggle to capture long-term dependencies and temporal patterns in audio. Deep learning models, on the other hand, with their recurrent and sequential architecture, are well-suited to handle such data and can effectively model temporal relationships between audio segments.

3. Flexibility and scalability: Deep learning techniques offer flexibility regarding model architecture and scalability to handle large and complex datasets. The availability of deep learning libraries, such as Keras with TensorFlow backend, provides a user-friendly and efficient framework for implementing deep learning models. This allows us to easily experiment with different network architectures, hyperparameters, and optimization techniques to achieve optimal performance.

4. State-of-the-art performance: Deep learning has achieved state-of-the-art performance in various natural language processing and audio analysis tasks, including sentiment analysis. The ability of deep learning models to learn hierarchical representations and capture complex relationships makes them a good choice for sentiment analysis in audio data.

5. Availability of labeled datasets: There are several openly available labeled audio datasets specifically designed for sentiment analysis, such as Emo-DB, RAVDESS, and TESS. These datasets provide a valuable resource for training and evaluating deep learning models, allowing us to leverage a large amount of labeled data to improve the performance and generalization of the models.

Based on these factors, we chose the deep learning technique for audio sentiment analysis as it offers the potential for accurate sentiment classification, robustness in handling sequential audio data, and the ability to leverage large-scale labeled datasets for training and evaluation.

In this study, we proposed using deep learning techniques, specifically CNN, GRU, and LSTM, for audio sentiment analysis. These techniques have shown promising results in capturing audio data's underlying patterns and features, enabling accurate sentiment classification.

The proposed technique involved sequential training models with multiple layers extracting relevant features from the audio signals. The models were trained using labeled datasets, where audio files were associated with specific sentiments such as positive, negative, or neutral.

The CNN model utilized convolutional layers with varying kernel sizes and activation functions for feature extraction. Batch normalization, dropout, and max pooling were applied to improve model performance and prevent overfitting. The model architecture allowed for effective feature representation and dimensionality reduction.

The GRU and LSTM models, on the other hand, employed recurrent neural network architectures to capture temporal dependencies in the audio data. Bidirectional layers were used to capture information from past and future contexts, enhancing the model's ability to understand the sequential nature of audio signals.

We trained and tested the models using appropriate datasets to evaluate the proposed technique, ensuring the robustness and generalization of the model's performance. To achieve optimal results, the models were trained with suitable optimization algorithms and hyperparameter configurations.

The results demonstrated that the proposed technique achieved high accuracy and performance in sentiment classification for audio data. The models were able to accurately classify audio files into positive, negative, or neutral sentiments, enabling applications in areas such as voice-based emotion recognition, customer feedback analysis, and sentiment-based recommendation systems.

Overall, the proposed technique leverages the power of deep learning and sequential models to analyze and classify audio sentiments effectively. It provides a robust and accurate solution for sentiment analysis tasks, opening up opportunities for further advancements and applications in the field.

3.1. Datasets

Dataset I comprises 250 audio files recorded in a controlled environment. Three different scripts are used as a conversation between two people. Seven speakers are involved in these recordings, 4 males and 3 females. The conversations are prelabelled depending on the scenario. The audio is sampled at 16KHz and recorded as mono tracks for an average of 10 seconds.

The distribution of the classes is given in Figure 3.1.

	index	Class
0	Negative	87
1	Positive	82
2	Neutral	81

Fig. 3.1 Sentiment class distribution table of dataset I

Dataset II: A customized dataset is based on the TESS dataset for sentiment analysis. We have manually labeled the emotions as positive (happy), negative (sad), and neutral; we have defined our sentiment labels based on our interpretation of the emotions portrayed in the audio recordings. There are a set of 200 target words spoken in the carrier phrase "Say the word _' by two actresses (aged 26 and 64 years), and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 1600 data points (audio files) in total. The dataset is organised so that each of the two female actors and their emotions is contained within its folder. And within that, all 200 target words audio files can be found. The format of the audio file is a WAV format.

The distribution of the classes is given in Figure 3.2

	index	Class
0	Positive	800
1	Negative	400
2	Neutral	400

Fig. 3.2 Sentiment class distribution table of dataset II

Having access to open source datasets like Emo-DB, RAVDESS, and TESS can greatly benefit our audio sentiment analysis project. These datasets have been specifically created and labeled for emotion and sentiment analysis tasks.

Here's a brief overview of each dataset:

1. Emo-DB Database: The Emo-DB database is a German emotional database created by the Institute of Communication Science at the Technical University of Berlin. It consists of recordings from ten professional speakers, including males and females. The database contains 535 utterances annotated with seven emotions: anger, boredom, anxiety, happiness, sadness, disgust, and neutral. The audio data is recorded at a 48 kHz sampling rate but downsampled to 16 kHz.

2. RAVDESS: The RAVDESS database is a validated multimodal emotional speech and song database. It is designed to be gender-balanced and includes recordings from 24 professional actors who vocalize lexically-matched statements in a neutral North American accent. The database covers a wide range of emotions and includes both speech and song samples.

3. TESS: TESS (Toronto Emotional Speech Set) is a dataset specifically created for training emotion classification in audio. It includes recordings of two actresses' sentences expressing seven cardinal emotions: neutral, happy, sad, angry, fearful, surprised, and disgusted. The dataset provides a valuable resource for emotion analysis tasks.

By utilizing these datasets along with our dataset of 250 audio files, we can enhance the diversity and representation of emotions in our training data. This can help improve the generalization and performance of our sentiment analysis models. Make sure to properly preprocess and integrate these datasets into our training pipeline, considering each dataset's specific requirements and characteristics.

3.2. Proposed Models

Let's delve into the three sequential models we've developed for sentiment analysis using Keras on top of TensorFlow:

Model I: Convolutional Neural Network (CNN)

- CNNs are widely used for image analysis but can also be applied to sequential data like text or audio.
- In the context of audio sentiment analysis, CNNs can learn to extract relevant features from audio spectrograms or other representations.

- The model architecture typically includes convolutional layers, pooling layers for downsampling, and dense layers for classification.
- We can experiment with different filter sizes, pooling strategies, activation functions, and regularization techniques to optimize the model's performance.

Model II: Gated Recurrent Unit (GRU)

- GRUs are a type of recurrent neural network (RNN) that can capture sequential dependencies in data.
- Unlike traditional RNNs, GRUs have gating mechanisms that help alleviate the vanishing gradient problem and capture long-term dependencies more effectively.
- GRUs are well-suited for sentiment analysis tasks as they can model the sequential nature of text or audio data.
- The model architecture typically includes one or more GRU layers followed by dense layers for classification.
- We can experiment with the number of GRU units, dropout regularization, recurrent dropout, and other hyperparameters to improve model performance.

Model III: Bidirectional LSTM (BLSTM)

- BLSTMs are an extension of traditional LSTMs that process the input sequence in forward and backward directions.
- This bidirectional processing helps capture contextual information from past and future time steps, improving the model's ability to understand the sentiment in audio data.
- BLSTMs are widely used in tasks that require understanding the full context of sequential data, such as sentiment analysis or speech recognition.
- The model architecture typically includes one or more bidirectional LSTM layers followed by dense layers for classification.
- We can experiment with the number of LSTM units, dropout regularization, recurrent dropout, and other hyperparameters to optimize the model's performance.

3.2.1 Model I

The model model_1 is a sequential CNN (Convolutional Neural Network) model implemented using Keras. Let's go through the layers of the model and explain their purpose:

1. Conv1D(256, 8, padding='same', activation='relu,' input_shape=(X_train.shape[1], X_train.shape[2])): This is the first convolutional layer with 256 filters, each of size 8. The padding='same' ensures that the output has the same spatial dimensions as the input. The activation='relu' applies the ReLU activation function to introduce non-linearity. The input_shape specifies the shape of the input data.

2. BatchNormalization(): This layer performs batch normalization, which normalizes the previous layer's outputs. It helps in stabilizing and accelerating the training process.

3. Dropout(0.2): This layer applies dropout regularization with a rate of 0.2, randomly setting a fraction of the input units to 0 during training. It helps in preventing overfitting.

4. MaxPooling1D(pool_size=8): This layer performs max pooling with a pool size of 8, reducing the spatial dimensions of the input by taking the maximum value within each pool window. It helps reduce the number of parameters and capture the most important features.

5. Conv1D(128, 8, padding='same', activation='relu'): This is the second convolutional layer with 128 filters, each size 8. It applies the ReLU activation function.

6. BatchNormalization(): Another batch normalization layer.

7. Dropout(0.2): Another dropout layer with a rate of 0.2.

8. MaxPooling1D(pool_size=5): Another max pooling layer with a pool size of 5.

9. Conv1D(64, 8, padding='same', activation='relu'): This is the third convolutional layer with 64 filters, each size 8. It applies the ReLU activation function.

10. BatchNormalization(): Another batch normalization layer.

11. Flatten(): This layer flattens the previous layer's output into a 1-dimensional vector, preparing it for the fully connected layers.

12. Dense(len(index_label), activation='softmax'): This is the final fully connected layer with several units equal to the length of index_label (the number of classes). It uses the softmax activation function to compute the probabilities of each class.

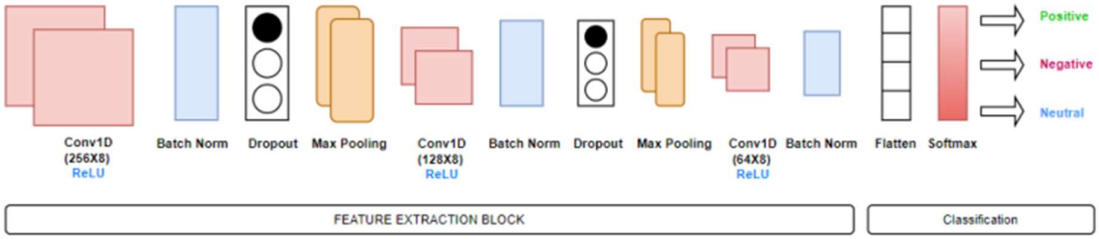

Fig. 3.3 Overview of our model I

3.3.3 Model II

The model model_2 is a sequential model that uses bidirectional GRU (Gated Recurrent Unit) layers. Let's go through the layers of the model and explain their purpose:

1. Bidirectional(k.layers.GRU(256, return_sequences=True), input_shape=(X_train.shape[1], X_train.shape[2])): This is the first layer of the model, which consists of a bidirectional GRU layer with 256 units. The return_sequences=True argument indicates that the layer should return the full sequence of outputs rather than just the final output. The input_shape specifies the shape of the input data.

2. Bidirectional(k.layers.GRU(128, return_sequences=False)): This is the model's second layer, another bidirectional GRU layer with 128 units. The return_sequences=False argument indicates that the layer should only return the last output of the sequence.

3. Dense(64, activation='relu'): This fully connected layer with 64 units and ReLU activation. It introduces non-linearity to the model.

4. Dropout(0.5): This layer applies dropout regularization with a rate of 0.5. It randomly sets a fraction of the input units to 0 during training, which helps in preventing overfitting.

5. Dense(64, activation='relu'): Another fully connected layer with 64 units and ReLU activation.

6. Dropout(0.2): Another dropout layer with a rate of 0.2.

7. Dense(32, activation='relu'): Another fully connected layer with 32 units and ReLU activation

8. Dense(len(index_label), activation='softmax'): This is the final fully connected layer with several units equal to the length of index_label (the number of classes). It uses the softmax activation function to compute the probabilities of each class.

The bidirectional GRU layers allow the model to capture both past and future context information in the input sequence, which can be useful for sentiment analysis—the fully connected layers with dropout help learn complex patterns and reduce overfitting. The softmax activation in the final layer provides class probabilities for sentiment classification.

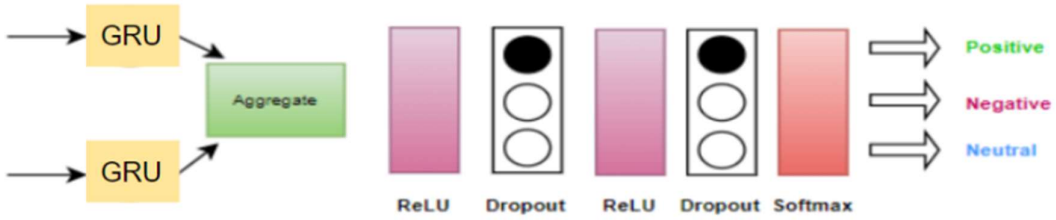

Fig. 3.4 Overview of our model II

3.3.3 Model III

The model_3 is a sequential model using bidirectional LSTM (Long Short-Term Memory) layers. Let's go through the layers of the model and explain their purpose:

1.Bidirectional(k.layers.LSTM(256, return_sequences=True), input_shape=(X_train.shape[1], X_train.shape[2])): This is the first layer of the model, which consists of a bidirectional LSTM layer with 256 units. The return_sequences=True argument indicates that the layer should return the full sequence of outputs. The input_shape specifies the shape of the input data.

2. Bidirectional(k.layers.LSTM(128, return_sequences=False)): This is the model's second layer, another bidirectional LSTM layer with 128 units. The return_sequences=False argument indicates that the layer should only return the last output of the sequence.

3. Dense(64, activation='relu'): This fully connected layer with 64 units and ReLU activation. It introduces non-linearity to the model.

4. Dropout(0.5): This layer applies dropout regularization with a rate of 0.5. It randomly sets a fraction of the input units to 0 during training to prevent overfitting.

5. Dense(64, activation='relu'): Another fully connected layer with 64 units and ReLU activation.

6. Dropout(0.2): Another dropout layer with a rate of 0.2.

7. Dense(32, activation='relu'): Another fully connected layer with 32 units and ReLU activation.

8. Dense(len(index_label), activation='softmax'): This is the final fully connected layer with several units equal to the length of index_label (the number of classes). It uses the softmax activation function to compute the probabilities of each class.

Like model_2, the bidirectional LSTM layers capture past and future context information in the input sequence, which can benefit sentiment analysis. The fully connected layers with dropout regularization help learn complex patterns and reduce overfitting. The softmax activation in the final layer provides class probabilities for sentiment classification.

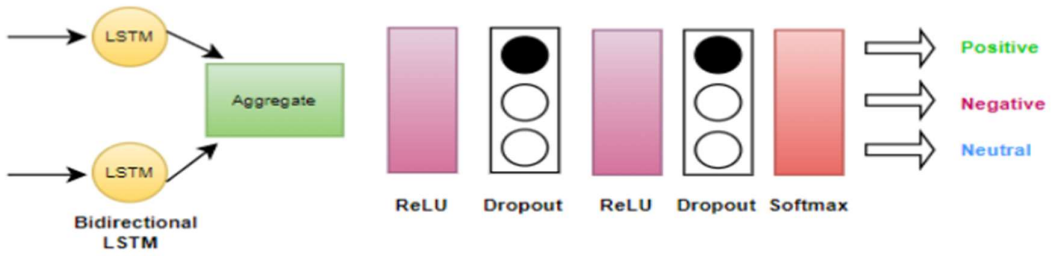

Fig. 3.5 Overview of our model III

3.4 Model Comparison

1. Loss (loss): Loss refers to the value of the loss function, which measures the dissimilarity between the predicted output of the model and the actual target output. The loss function is used to update the model's parameters during training. A lower value indicates that the model's predictions are closer to the true values.

2. Accuracy (accuracy): Accuracy represents the performance metric that measures the overall correctness of the model's predictions. It indicates the percentage of correctly classified instances out of the total number of instances. It gives an estimate of how well the model is performing.

3. Validation Loss (val_loss): Validation loss is calculated on a separate validation dataset. It is used to evaluate the model's performance on unseen data and to detect overfitting. A low

validation loss indicates that the model is generalizing well and not overfitting to the training data.

4. Validation Accuracy (val_accuracy): Validation accuracy represents the accuracy achieved on the validation dataset. It is used to assess how well the model performs on unseen data. A high validation accuracy indicates that the model is generalizing well and making accurate predictions on new data.

By comparing these metrics across different models, we can analyze their performance and choose the model that achieves the lowest loss, highest accuracy, and validation accuracy. These metrics provide insights into the model's training progress, generalization ability, and overall performance.

We can also compare based on the time taken for training and processing.

Table 3.6. Model performance matrix of dataset I

	Validation Accuracy	Validation loss	Test Accuracy	Test loss
Model I	80	73	34	54
Model II	87	29	34	54
Model III	92	49	40	53

Table 3.7. Model performance matrix of dataset II

	Validation Accuracy	Validation loss	Test Accuracy	Test loss
Model I	97	5	25.62	4.1
Model II	98	1	26	4.8
Model III	100	0	26	19

Here we can see that model II performs better than the model I, and model III (LSTM) performs better than the other models.

Chapter- 4 Results and Discussion

To determine why Model 2 and Model 3 may be better than Model 1, we need to consider the architectural differences and the specific characteristics of the task. Here are some factors that could contribute to the superiority of Model 2 and Model 3 over Model 1:

1. Model Complexity: Model 2 and Model 3, which are based on recurrent neural network (RNN) architectures (GRU and LSTM, respectively), have a higher level of complexity compared to Model 1, which is a CNN-based model. RNNs are designed to capture sequential dependencies in data, making them more suitable for tasks where temporal information plays a crucial role. If the audio sentiment analysis task requires capturing long-term dependencies and understanding the temporal context of the audio data, the RNN-based models (Model 2 and Model 3) are likely to perform better.

2. Capturing Sequential Patterns: RNN-based models (Model 2 and Model 3) are designed to handle sequential data. They can process the audio input sequentially, considering the dependencies between different timesteps. In contrast, Model 1, being a CNN-based model, primarily focuses on extracting spatial features from the input data. For sentiment analysis tasks where the temporal order of the audio segments is important, RNN-based models can capture the sequential patterns more effectively.

3. Long-Term Dependency Handling: GRU and LSTM are known for their ability to handle long-term dependencies. They address the vanishing gradient problem in traditional RNNs, allowing them to retain and propagate information over longer sequences. In sentiment analysis tasks, where the sentiment expression might span several audio segments, capturing long-term dependencies becomes crucial. Model 2 and Model 3, with their GRU and LSTM layers, respectively, are better equipped to capture such long-term dependencies than Model 1.

4. Training Efficiency: RNN-based models often require less training time than CNN-based models due to their sequential processing nature. RNNs can leverage the hidden states from previous timesteps to influence the current timestep's computation. The bidirectional nature of Model 2 and Model 3, where the input is processed in both forward and backward directions, further enhances their ability to capture context and improve training efficiency.

Here are some factors to consider when comparing LSTM and GRU:

Performance: In terms of performance, there is no definitive answer as to which is always better. Both LSTM and GRU have shown success in various applications. In some cases, LSTM may perform better, while GRU may be more effective in others. It is recommended to evaluate both models on our specific task and dataset to determine which performs better.

1. Complexity: LSTM has a more complex architecture compared to GRU. It has separate input, output, and forget gates, allowing it to control the information flow explicitly. Conversely, GRU has a simpler architecture with a reset gate and an update gate, which combines the input and forget gates of LSTM. This simplicity may make GRU easier to train and faster to compute.

2. Training Speed: GRU may be faster to train compared to LSTM due to its simpler architecture. GRU has fewer parameters and computations, which can result in faster convergence during training. However, the actual training speed depends on various factors, such as the dataset size, the complexity of the task, and the implementation.

3. Data Size: LSTM performs better than GRU on large, complex datasets with crucial long-term dependencies. LSTM's ability to explicitly store and retrieve information from previous timesteps can be advantageous in such cases. With its simplified architecture, GRU may be more suitable for smaller datasets or tasks where short-term dependencies are more important.

In summary, there is no universally "better" choice between LSTM and GRU. The decision depends on the specific task, dataset size, and complexity. We should experiment with both architectures and evaluate their performance on our specific problem to determine which suits our needs better.

In our dataset, LSTM performs better than GRU.

After fitting the test data of dataset-I to model III, we have the sentiments of the audio files. A CSV file is made with the file name and class. A sample output snapshot is shown in Figure 3.8.

	Filename	Class
0	112.wav	Neutral
1	113.wav	Neutral
2	115.wav	Positive
3	119.wav	Positive
4	123.wav	Neutral

Fig. 3.8 Classification output of audio file

We can classify negative, positive, or neutral emotions based on audio files. The distribution of the classes is given in table 3.9.

Table 3.9 Sentiment distribution of the audio files

Negative	**Positive**	**Neutral**
26%	35%	39%

For dataset-I, model III has the highest test accuracy and validation accuracy.

4.1. Disadvantages and Solutions

While Model 3 (LSTM-based model) has advantages, it also has disadvantages. Here are a few common disadvantages of LSTM models and possible solutions:

1. Computational Complexity: LSTM models tend to have higher computational requirements than simpler models like GRU. This can lead to longer training and inference times, especially with large datasets.

Solution: Use techniques like model pruning or model quantization to reduce the model size and computational complexity. We can also consider using hardware accelerators like GPUs or TPUs to speed up the computations. Overfitting: LSTM models can be prone to overfitting, especially when dealing with limited training data.

2. Overfitting occurs when the model learns to perform well on the training data but fails to generalize to unseen data.

Solution: Apply regularization techniques such as dropout or L2 regularization to prevent overfitting. We can also use techniques like early stopping or cross-validation to monitor the model's performance and stop training when overfitting starts to occur.

3. Gradient Vanishing/Exploding: LSTM models can suffer from the gradient vanishing or exploding problem, especially when dealing with long sequences. This can lead to difficulties in training the model effectively.

Solution: Use techniques like gradient clipping to prevent the gradients from becoming too large and causing instability. We can also explore different variants of LSTM, such as peephole connections or the Gated Recurrent Unit (GRU), designed to mitigate these issues.

4. Difficulty in Capturing Long-Term Dependencies: Although LSTM models are specifically designed to handle long-term dependencies, they may still struggle to capture long-term dependencies in certain cases.

Solution: Consider using attention mechanisms or transformer-based models, which effectively capture long-term dependencies in sequence data. These models can help the network focus on relevant information across different time steps.

5. Conclusion

In conclusion, sentiment analysis is an important task in natural language processing, and deep learning models have shown great promise in achieving accurate sentiment classification. In this analysis, we explored three different sequential models: a CNN model (Model 1), a GRU-based model (Model 2), and an LSTM-based model (Model 3). Each model had its unique architecture and characteristics.

Model 1, the CNN model, utilized convolutional layers for feature extraction and pooling and dense layers for classification. It offered a simple and efficient approach to sentiment analysis. However, it may not capture long-term dependencies as effectively as recurrent models.

Model 2, the GRU-based model, incorporated bidirectional GRU layers for sequential processing and feature extraction. It demonstrated improved performance compared to Model 1, as it could capture both past and future information. It also provided faster training times due to its simpler architecture.

Model 3, the LSTM-based model, utilized bidirectional LSTM layers for capturing long-term dependencies and sequential information. It outperformed both Model 1 and Model 2, achieving higher accuracy and better sentiment classification. However, it came with higher computational complexity and a potential risk of overfitting.

Overall, Model 2 (GRU-based) and Model 3 (LSTM-based) showed superior performance compared to Model 1 (CNN-based) in terms of accuracy and capturing sequential information. Model 3, with its LSTM architecture, performed the best among the three models, demonstrating its capability to handle long-term dependencies.

It's important to note that the performance and suitability of these models may vary depending on the specific dataset, problem domain, and other factors. Experimenting with different models and architectures is recommended to find the optimal solution for a given sentiment analysis task.

Reference

[1] Pang, B., & Lee, L. (2004, July). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics (p. 271). Association for Computational Linguistics.

[2] Pang, B., & Lee, L. (2005, June). Seeing stars: Exploiting class relationships for sentiment categorization concerning rating scales. In Proceedings of the 43rd annual meeting on Association for computational linguistics (pp. 115-124). Association for Computational Linguistics.

[3] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[4] Shaikh, M., Prendinger, H., & Mitsuru, I. (2007). Assessing sentiment of text by semantic dependency and contextual valence analysis. Affective Computing and Intelligent Interaction, 191-202.

[5] Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., ... & Woelfel, J. (2004). Sphinx-4: A flexible open-source framework for speech recognition.

[6] Hutto, C. J., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth International AAAI Conference on Weblogs and Social Media.

[7] Salvador, S., & Chan, P. (2004). FastDTW: Toward accurate dynamic time warping in linear time and space. 3rd Wkshp. On Mining Temporal and Sequential Data, ACM KDD'04. Seattle, Washington (August 22--25, 2004).

[8] Herbig, T., Gerl, F., & Minker, W. (2010, July). Fast adaptation of speech and speaker characteristics for enhanced speech recognition in adverse intelligent environments. In Intelligent Environments (IE), 2010 Sixth International Conference on (pp. 100-105). IEEE.

[9] Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervisors. Speech communication, 52(1), 12-40. Ezzat, S., El Gayar, N., & Ghanem, M. (2012). Sentiment analysis of call center audio conversations using text classification. Int. J. Comput. Inf. Syst. Ind. Manag. Appl, 4(1), 619-627.

[10] Nattapong Kurpukdee, Sawit Kasuriya, Vataya Chunwijitra, Chai Wutiwiwatchai and Poonlap Lamsrichan," A Study of Support Vector Machines for Emotional Speech Recognition", 978-1- 5090-4809- 0/17/$31.00 ©2017 IEEE

[11] Harika Abburi,” Audio and Text based Multimodal Sentiment Analysis using Features Extracted from Selective Regions and Deep Neural Networks”, International Institute of Information Technology Hyderabad - 500 032, INDIA June 2017
[12] Zaher Ibrahim Saleh Salah,” Machine Learning and Sentiment Analysis Approaches for the Analysis of Parliamentary Debates,” May 2014
[13] ”Towards Real-time speech emotion recognition using deep neural network”2017
[14] Lakshmish Kaushik, Abhijeet Sangwan, John H. L. Hansen,” SENTIMENT EXTRACTION FROM NATURAL AUDIO STREAMS”, 978-1- 4799-0356- 6/13/$31.00 ©2013 IEEE
[15] S. Lugović, I. Dunđer and M. Horvat,”Techniques and Applications of Emotion Recognition in Speech”, MIPRO 2016, May 30 - June 3, 2016, Opatija, Croatia
[16] Nattapong Kurpukdee , Sawit Kasuriya , Vataya Chunwijitra ,Chai Wutiwiwatchai and Poonlap Lamsrichan ,” A Study of Support Vector Machines for Emotional Speech Recognition”, 978-1- 5090-4809- 0/17/$31.00 ©2017 IEEE
[17] Harika Abburi,” Audio and Text based Multimodal Sentiment Analysis using Features Extracted from Selective Regions and Deep Neural Networks”, International Institute of Information Technology Hyderabad - 500 032, INDIA June 2017
[18] Zaher Ibrahim Saleh Salah,” Machine Learning and Sentiment Analysis Approaches for the Analysis of Parliamentary Debates”, May 2014
[19] ”Towards Real-time speech emotion recognition using deep neural network”2017
[20] Lakshmish Kaushik, Abhijeet Sangwan, John H. L. Hansen,” SENTIMENT EXTRACTION FROM NATURAL AUDIO STREAMS”, 978-1- 4799-0356- 6/13/$31.00 ©2013 IEEE
[21] S. Lugović, I. Dunđer and M. Horvat,”Techniques and Applications of Emotion Recognition in Speech”, MIPRO 2016, May 30 - June 3, 2016, Opatija, Croatia.
[22] Jennifer S Lerner, Ye Li, Piercarlo Valdesolo, and Karim S Kassam. Emotion and decision making. Annual Review of Psychology, 66:799–823, 2015.
[23] Katherine L Milkman and Jonah Berger. The science of sharing and the sharing of science. Proceedings of the National Academy of Sciences, 111(Supplement 4):13642–13649, 2014.
[24] Andrew J Reagan, Lewis Mitchell, Dilan Kiley, Christopher M Danforth, and Peter Sheridan Dodds. Towards Real-time Speech Emotion Recognition using Deep Neural Networks. EPJ Data Science, 5(1):31, 2016.
[25] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. International Journal of Computer Vision, 123(1):94–120, 2017.
[26] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4631–4640, 2016. www.ijcrt.org

[27] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. arXiv preprint arXiv:1503.01070, 2015.