# Multimodal Speech Emotion Recognition

by

Marina Kreplak

Master in Artificial Intelligence
June 22, 2020

*Tutor:*
Lluís Padró Cirera

*Advisor:*
Mohammed Adil Moujahid

Facultat de
Matemàtiques

Escola Tècnica
Superior d'Enginyeria

Facultat d'Informàtica
de Barcelona (FIB)

# Abstract

The recognition of emotions in speech is one of the most challenging topics in data science. In this work, we define a pipeline for the study of multimodal speech recognition, using a wide set of features from audio samples and text transcripts.
This work aims to study the interaction and contribution of multimodal features and for this purpose, three types of features have been selected. We extract a set of handcrafted features related to speech prosody, along with classical mel spectrogram acoustic features and TF-IDF for text. Combining these three types of data we evaluate the contribution that they represent to each other.
This Thesis also provides a comparative study between the classical machine learning models performance over neural architectures in terms of performance and learning potential from speech. Finally, it presents an application that provides emotion classification and feedback retrieval for misclassified samples.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Despite the great progress made in artificial intelligence in the last decade, we are still far from being able to naturally interact with machines. There is still a big gap between natural human interactions and human-machine interactions, and this is partly because machines do not understand our emotional states and therefore they have no way to react to them.
This Thesis presents a comparative study and an effective approach to address the Emotion Recognition in Speech (SER) task. The following section provides an introduction to this thesis, organized as follows: in section 1.1 we introduce the reasons and needs of this work. In section 1.2 we present the main problems and challenges to be faced. Thereafter in section 1.3 we summarize the goals to be achieved by this work. And finally, in section 1.4 we describe the structure of this Thesis.

## 1.1 Motivation

Communication is the key to human existence, and human interactions lead frequently and undeniably to ambiguous situations. For instance, the sentence "This is great." could be said under either exciting or angry emotional states. Humans are able to resolve ambiguity in most cases because we can efficiently comprehend information from multiple domains such as speech, words and images.

An important application is human-computer interaction, typically in the context of conversational agents. Users of agents such as Siri or Google Assistant will attest that these systems lack relatability and fail to elicit empathy from the user. One way to improve the relatability of such systems is to give them the capacity to detect emotion from speech, allowing the system to respond in a more appropriate manner.

In developing emotionally aware intelligence, the very first step is building ro-

bust emotion classifiers that display good performance regardless of the application. In particular, the speech emotion recognition task (SER) is one of the most important problems in the field of para-linguistics. This field has recently broadened its applications, as it is a crucial factor in optimal human-computer interactions, including dialog systems.
The goal of speech emotion recognition is to predict the emotional content of speech and to classify speech according to one of several labels (i.e., happy, sad, neutral, and angry). This emotional prediction can be made using only the features contained in our speech, as well as with the semantic meaning of our words. The combination of both is known as multi-modal speech emotion recognition (MSER).

MSER basically consists in two steps: feature extraction from text and speech, and emotion classification. During the last years there's been an extensive research on both matters, and a large variety of solutions have been proposed. On one hand, Feature Extraction for SER remains a challenging task, while Emotion Classification presents a wide variety of possible solutions where there is plenty room for improvement.

This Master Thesis aims to address the emotion recognition in speech problem, and propose a solution based on speech and text by comparing different state-of-the-art architecture approaches to increase the overall accuracy achieved in previous works.

## 1.2 Main Challenges

The goal of speech emotion recognition is to predict the emotional content of speech and to classify speech according to one of several labels. Various types of deep learning methods have been applied to increase the performance of emotion classifiers; however, this task is still considered to be challenging for several reasons.

First, insufficient data for training complex neural network-based models are available, due to the costs associated with human involvement. There are just a few datasets available for tasks such as SER, and in most cases they are composed of a relatively small amount of samples. The emotion annotations require a very complex process since the nature of emotions is very subjective.

Variation in voice tones as well as internal physiological changes while uttering a sentence (or even a single word) combine to generate the speaker's emotional state. Perfect recognition of emotions is not easy even by humans when listening to each other; sometimes the human cannot recognize his own innermost emotion [Al-Talabani et al., 2015]. Categorizing emotional speech samples is a serious challenge due to the long debate about the real meaning of "emotion" and the emotional classes that should be dealt with.

Another challenge is to identify emotion relevant features that can be extracted from the raw speech signal or its frequency domain version. Recent studies observed that emotion related information in the speech is spread along different kind of features. This could be due to the acoustic variability as a consequent of the existence of different sentences, speakers, speaking styles, and speaking rates

## 1.3 Goals of this work

Considering the challenges observed in the domain of this thesis, this work is conceived with the following goals:

1. Define a common pipeline for Speech Emotion Recognition that combines acoustic and textual data. This means defining which are the best type of features and model architectures for a multimodal approach.
2. Provide a comparative study between the performances of 1) lighter machine learning and 2) deep learning based models.

## 1.4 Structure of this Thesis

This Thesis is structured as follows:

- In chapter 2 we explain the background context of Speech Emotion Recognition, the related work and the current State-of-the-art.
- In chapter 3 we explain the Methodology followed in this work and the approaches taken to fulfill our goals
- In chapter 4 we explain the settings for the experiments taken
- In chapter 5, the obtained results are presented and explained
- And at last, in chapter 6 we explain our conclusions and improvements we could include in a future work

# Chapter 2

# Background and related work

## 2.1 Dataset

One of the most interesting para-linguistic messages expressed during human interaction is the emotional state of the subjects, which is conveyed through both speech and gestures. The tone and energy of the speech, facial expressions, torso posture, head position, hand gestures, and gaze are all combined in a non-trivial manner, as they unfold during natural human communication. Even if we consider only one of these non-verbal feature (such as the voice tone), much more robust models can be developed and implemented.

### 2.1.1 Main limitations in existing databases

In this context, one of the major limitations in the study of emotion expression is the lack of databases with genuine interaction that comprise integrated information from most of these channels. [Douglas-Cowie et al., 2003] made an analysis of several existing emotional databases. They concluded that in most of the databases, the subjects were usually asked to ”act” or ”simulate” emotions while being recorded. While desirable from the viewpoint of providing controlled elicitation, these simplifications in data collection could discard important information observed in real life scenarios.
As a result, the performance of emotion recognition significantly decreases when the automatic recognition models developed by such databases are used in real life applications, where a blend of emotions is observed [Devillers et al., 2005].

Another limitation of existing corpora is that the recorded materials often consist of isolated utterances or dialogues with few turns [Douglas-Cowie et al., 2003]. This setting neglects important effects of contextualization, which play a crucial role in how we perceive and express emotions. Likewise, most of the existing databases contain only the acoustic speech channel. Therefore, these corpora cannot be used to study the information that is conveyed through the other communication channels.

Other limitations of current emotional databases are the limited number of subjects, and the small size of the databases. Similar observations were also presented in the review presented by [Ververidis and Kotropoulos, 2012].

### 2.1.2 IEMOCAP Database

Considering the limitations presented in the previous section, the database chosen for this work has been the interactive emotional dyadic motion capture database (IEMOCAP).

In this database ten actors were recorded in dyadic sessions (5 sessions with 2 subjects each). They were asked to perform three selected scripts with clear emotional content. In addition to the scripts, the subjects were also asked to improvise dialogues in hypothetical scenarios, designed to elicit specific emotions (happiness, anger, sadness, frustration and neutral state). One participant of the pair was motion captured at a time during each interaction.

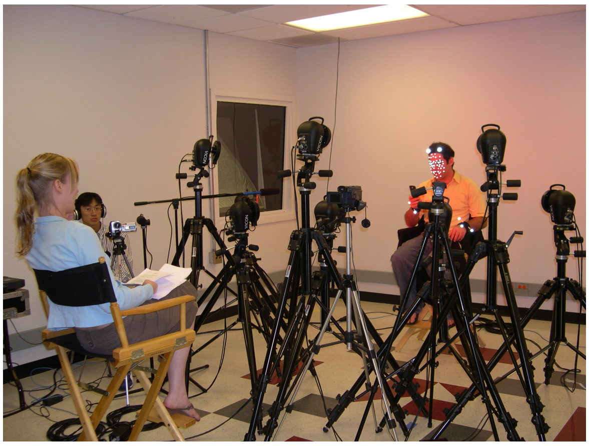


Figure 2.1: IEMOCAP recording session. *Source: sail.usc.edu*

Also, fifty-three facial markers were attached to the subject being motion captured, who also wore wristbands and a headband with markers to capture hand and head motion, respectively. Using this setting, the emotions were elicited within a proper context, improving the authenticity of the captured emotional data. Furthermore, gathering data from ten different subjects increases the plausibility of effectively analyzing trends observed in this database on a more general level. In total, the database contains approximately twelve hours of data.

Each utterance in the dataset was annotated by at least 3 human annotators, and besides the categorical emotion attributes, three dimensional attributes were also provided: valence, activation and dominance.

## 2.2 Representation of emotional models

Due to the multi-disciplinary nature of research on emotions, different representation schemes and models have emerged hampering comparison across different approaches.

In NLP-oriented sentiment and emotion analysis, the most popular representation scheme is based on semantic polarity, the positiveness or negativeness of a word or a sentence, while slightly more sophisticated schemes include a neutral class or even rely on a multi-point polarity scale. From an NLP point of view, those can be broadly subdivided into **categorical and dimensional models**.

### 2.2.1 Categorical models

Categorical models assume a small number of distinct emotional classes (such as Anger, Happiness, Fear or Sadness) that all human beings are supposed to share [Buechel and Hahn, 2017]. Although the view that some emotions are more "basic" than others is widely accepted by emotion theorists, there is little agreement on which emotions should be included in the list of the basic ones. Their number varies depending on the theory.

The most popular list, sometimes referred to as "The Big Six", was used by [Ekman et al., 1969] in their research on universal recognition of emotion from facial expression. Ekman's six basic emotions are:

- Happiness
- Surprise
- Fear
- Anger
- Disgust
- Sadness

These emotions are still the most commonly accepted candidates for basic emotions. All Ekman's six basic emotions are included in IEMOCAP, along with **excitement, frustration and neutral** emotions.

### 2.2.2 Dimensional models

On the other hand, **dimensional models** are centered around the notion of compositionality. They assume that emotional states can be best described as a combination of several fundamental factors, i.e., emotional dimensions.

One of the most popular dimensional models is the Valence-Arousal-Dominance (VAD) model, proposed by [Bradley and Lang, 1994] which postulates three orthogonal dimensions, namely **Valence** (corresponding to the concept of polarity), **Arousal** or activation (a calm-excited scale) and **Dominance** (perceived degree of control in a social situation).

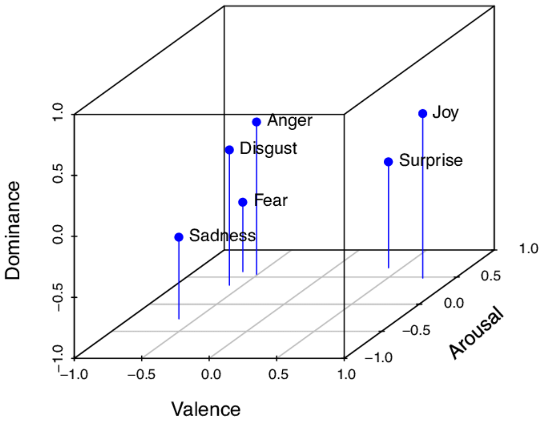


Figure 2.2: The emotional space spanned by the Valence-Arousal-Dominance model representing Ekman's six basic emotions *Source: [Buechel and Hahn, 2017]*

When one of these dimensional models is selected, the task of emotion analysis is most often interpreted as a regression problem (predicting real-valued scores for each of the dimensions). On the other hand, a categorical representation makes the emotion analysis task a classification problem.
In this work, we will use the categorical annotations provided in IEMOCAP for each of its utterances. Therefore, we will adopt the categorical model for emotions and treat the Speech Emotion Recognition task as a classification problem.

## 2.3 State-of-the-art

Classical machine learning algorithms, such as hidden Markov models (HMMs), support vector machines (SVMs), and decision tree-based methods, have been employed in speech emotion recognition problems [Seehapoch and Wongthanavasu, 2013]. Recently, researchers have proposed various neural network-based architectures to improve the performance of speech emotion recognition. An initial study utilized deep neural networks (DNNs) to extract high-level features from raw audio data and demonstrated its effectiveness in speech emotion recognition.[Han et al., 2014]

With the advancement of deep learning methods, more complex neural based architectures have been proposed. Convolutional neural network (CNN)-based models have been trained on information derived from raw audio signals using spectrograms or audio features such as Mel-frequency cepstral coefficients (MFCCs) and low-level descriptors (LLDs) [Badshah et al., 2017]. These neural network-based models are combined to produce higher-complexity models and achieved the best-recorded performance when applied to the IEMOCAP dataset.

Another line of research has focused on adopting variant machine learning techniques combined with neural network based models. One researcher utilized the multi-object learning approach and used gender and naturalness as auxiliary tasks so that the neural network-based model learned more features from a given dataset [Kim et al., 2017]. Another researcher investigated transfer learning methods, leveraging external data from related domains [Gideon et al., 2017].

As emotional dialogue is composed of sound and spoken content, researchers have also investigated the combination of acoustic features and language information, built belief network-based methods of identifying emotional key phrases, and assessed the emotional salience of verbal cues from both phoneme sequences and words [Schuller et al., 2004] and [Zhang et al., 2019]. However, [Yoon et al., 2018] seems to be the first to use information from speech signals and text sequences simultaneously in an end-to-end learning neural network-based model to classify emotions.

Since we will be using IEMOCAP dataset, in the next subsections we are going to focus on the state-of-the-art results achieved when using this dataset.

### 2.3.1 Deep Dual Recurrent Encoder Approach

In 2018, [Yoon et al., 2018] proposed in his paper a novel deep dual recurrent encoder model that simultaneously utilizes audio and text data in recognizing emotions from speech. Using IEMOCAP, they extracted MFCC features from audio and used the GloVe word embedding for textual feature extraction. They classified 4 emotions: Happy, Angry, Sad and Neutral.

As shown in figure 2.3, audio signals and textual information were encoded by using separate recurrent encoders (ARE and TRE) to create a Multimodal Dual Recurrent Encoder (MDRE). In their approach they also experimented with adding an attention layer and creating a MDRE with attention (MDREA).

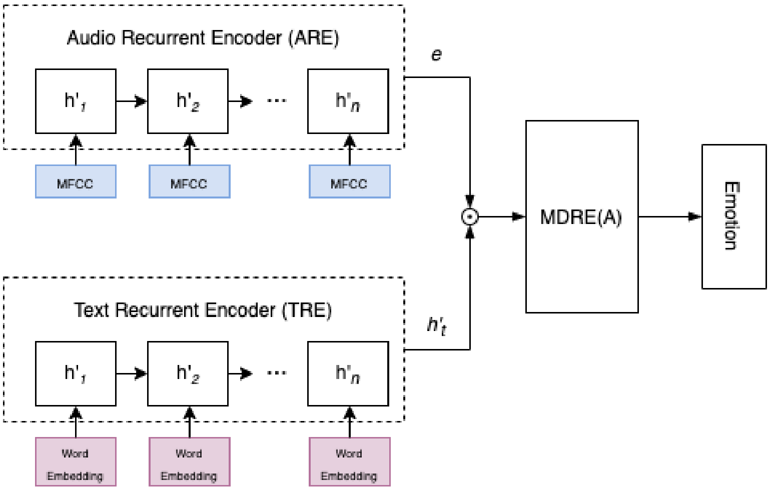


Figure 2.3: Multimodal dual recurrent encoder architecture

In this last experiment, the attention weight $\mathbf{a_t}$ was calculated as the dot product of the last hidden state $\mathbf{h'_t}$ of the text-RNN and the final encoding vector of the audio-RNN. However, the MDREA model did not match the performance of the MDRE model, even though it utilized a more complex architecture.

They achieved a maximum weighted accuracy of 0.718 with the MDRE model:

| Model | WAP |
|---|---|
| ARE | $0.546 \pm 0.009$ |
| TRE | $0.635 \pm 0.018$ |
| **MDRE** | $\mathbf{0.718 \pm 0.019}$ |
| MDREA | $0.690 \pm 0.019$ |

Table 2.1: Results achieved by Yoon et al. on IEMOCAP.

### 2.3.2 Hand-crafted features for Speech Classification

In 2019, [Sahu, 2019] developed a series of hand-crafted features from the audio signal, different than the traditional MFCC or MEL spectrogram, also using IEMOCAP. In this study, they extracted audio features such as **pitch, harmonics and pause** applying some feature engineering.

This study was focused on comparing the performance of these features when using different model architectures. As shown in Figure 2.4 they compared a series of Machine Learning and Deep Learning models among which are an LSTM, a Multi-Layer Perceptron an XGBoost classifier and a Random Forest classifier. They classifier 6 different emotions instead of 4, but applying pretty unsophisticated data augmentation techniques.

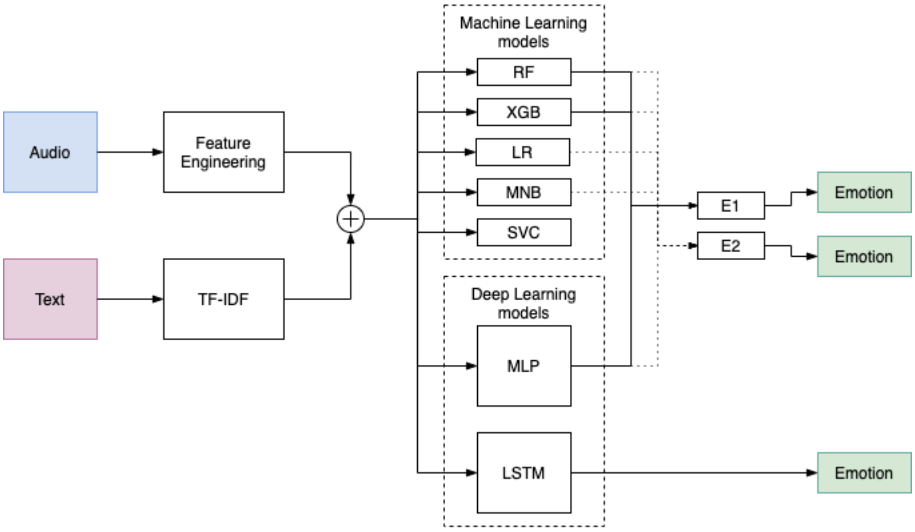


Figure 2.4: ML and DL models for multimodal SER with hand-crafted features

[Sahu, 2019] observed that assembling multiple ML models (E1, E2) led to some improvement in the performance. E1 combined RF, XGB and MLP and E2 ensembled RF, XGB, MLP, MNB and LR. They achieved a maximum accuracy of 0.703 with one of their assembled models (see Table 2.2).

| Model | Accuracy | F1-score | Precision | Recall |
|---|---|---|---|---|
| RF | 65.3 | 65.8 | 69.3 | 65.5 |
| XGB | 62.2 | 63.1 | 67.9 | 61.7 |
| SVM | 63.4 | 63.8 | 63.1 | 65.6 |
| MNB | 60.5 | 60.3 | 70.3 | 57.1 |
| MLP | 66.1 | 68.1 | 68.0 | 69.6 |
| LR | 63.2 | 63.7 | 66.9 | 62.3 |
| LSTM | 64.3 | 64.7 | 66.1 | 65.0 |
| E1 | **70.3** | 67.5 | **73.2** | 65.5 |
| **E2** | 70.1 | **71.8** | 72.9 | **71.5** |

Table 2.2: Results achieved by [Sahu, 2019] on IEMOCAP.

### 2.3.3 Self Attention Mechanism approach

Another line of research has focused on Attention based techniques. In 2019 [Li et al., 2019] proposed an attention mechanism for machine translation and speech recognition, achieving great results **but using only audio features**.

The major contributions of this work were **classifying emotion using spectrogram based self-attentional CNN-BLSTM model** and combining **emotion classification and gender classification** using multitask learning. This means that they incorporated Gender classification to consider the relationship between the two tasks (Gender and Emotion classification) and better classify emotion in a multitask learning manner.

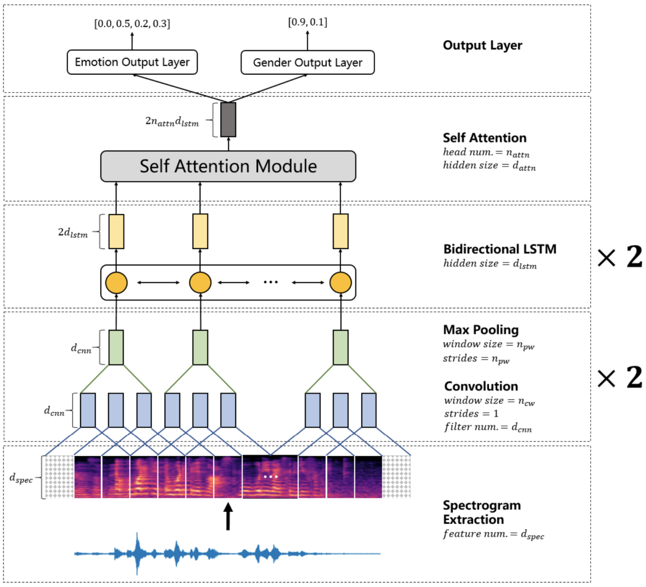


Figure 2.5: Self-Attentional CNN BLSTM model for emotion and gender classification. *Source: [Li et al., 2019]*

In this approach, samples are extracted using a STFT and the calculated spectrogram is mapped to Mel scale. These samples feed the Neural Network of the system which integrates a convolutional layer and a max pooling layer, a bidirectional LSTM and finally a self attention layer.

They achieved great results for the multitask learning task (considering the classification of gender too), and proved that the self-attentional component in their architecture improved the state-of-the-art results in SER. Their results are represented in Table 2.3:

| Method | WA | UA |
|---|---|---|
| **Full model** | **81.6** | **82.8** |
| Self-attention | 55.3 | 51.1 |
| Multitask learning | 70.5 | 72.6 |

Table 2.3: Results achieved by Li et al. on IEMOCAP

[Li et al., 2019] reached new state-of-the-art results on the emotion classification combining it with gender, although the accuracy of gender classification individually was not as good as common gender classification results. This could be because IEMOCAP is especially collected for emotion recognition research and not suitable for gender recognition.

### 2.3.4 Summary

[Yoon et al., 2018] achieved great results using a Multimodal Dual Recurrent Encoder architecture, but using only MFCC as audio features. On the other hand, [Sahu, 2019] also achieved good results using handcrafted audio features but using common ML classifiers. Finally, [Li et al., 2019] used a Self-Attentional CNN architecture and achieved great results, but only when classifying emotion together with gender, and using only audio data as input and not text.

| | **Input Data** | **Features** | **Architecture** | **Year** | **Dataset** | **Best Score (accuracy)** |
|---|---|---|---|---|---|---|
| **Yoon et al.** | Text + Audio | MFCC | MDRE | 2018 | IEMOCAP | 71.8 |
| **Sahu** | Text + Audio | Handcrafted features | Traditional ML models | 2019 | IEMOCAP | 70.3 |
| **Li et al.** | Audio only | Mel Spectrogram | CNN-BLSTM | 2019 | IEMOCAP | 81.6 |

Table 2.5: State of the art summary

# Chapter 3

# Methodology

The analysis on the current State-of-the-art for SER establish a wide variety of methods regarding the features, architectures and type of data used in every approach. But most of all, it determines that there is no consolidated methodology for a common multimodal approach for SER.

[Li et al., 2019] proposed the best performing architecture (CNN-BLSTM) for SER but with no feature engineering methods, while [Sahu, 2019] defined a set of handcrafted features suitable for the study of SER. In this Chapter we will define a standard approach for SER by combining these two approaches into an end-to-end pipeline that: 1) uses audio and text data as input, 2) performs feature engineering to produce handcrafted features and 3) uses a deep neural approach architecture to classify emotions.

## 3.1 Our proposal

The pipeline proposed in this Thesis is based on five main stages: Data Processing, Feature Extraction, Model Training, Model Evaluation and Emotion Classification.

The goal of the pipeline presented in Figure 3.1 is the multi-modal emotion classification of a human speech audio signal. To do so, a classifier needs to be trained with Machine Learning/Deep Learning techniques, and audio and text data are needed for training. Both types of training data are extracted from the same dataset (IEMOCAP). The final emotion classification is performed over a non-supervised sample.

The main addressed research problem has been the performance study of different **audio features** and which type of architecture worked best for each one. Another question for analysis has been the **interaction between text and**

**audio features and which combination of features** resulted better.

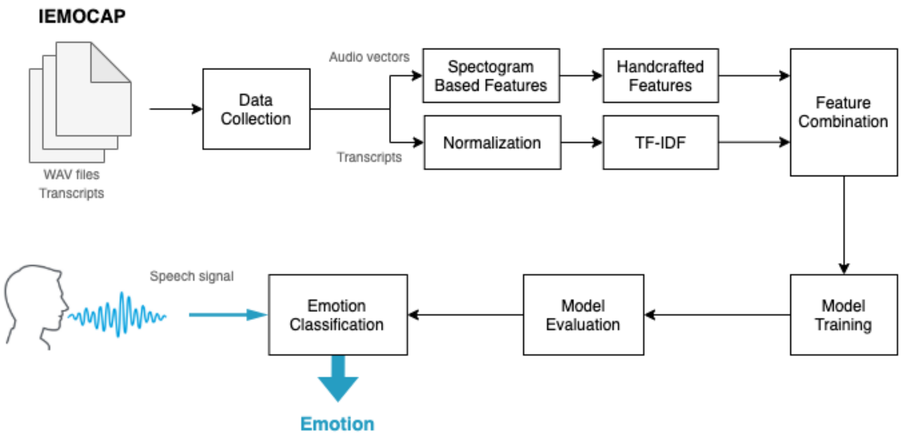


Figure 3.1: General approach pipeline

Another issue considered for analysis has been the performance of several classifier models with different architectures. Several different types of models have been optimized and compared in order to conclude with which of them we could achieve the closest results to the state-of-the-art approaches.

The final step in this pipeline is the emotion recognition from an **unseen audio sample** by using the best performing trained model. Since we will need the audio sample along with its text, the transcript of the audio sample will be extracted using Google Cloud Speech API.

## 3.2 Data Collection

The very first step in our work has been collecting the needed data from IEMOCAP and structuring it to be studied. As described in Section 2.1, IEMOCAP provides video files besides the audio files and transcripts corresponding to each utterance. In this initial stage we have retrieved only text and audio files and structured it into our own file directory.

The emotion annotation for each utterance contains a categorical value for 11 different emotions and three numerical values comprised between 0 and 5, corresponding to valence, arousal and dominance attributes. As we have approached this work to be a classification problem, we have only extracted the categorical value for its study. The initial distribution of samples and emotions is represented in Table 3.1.

| **Emotion** | **Number of samples** |
|---|---|
| Anger | 1103 |
| Sadness | 1084 |
| Happiness | 595 |
| Neutral | 1708 |
| Excitement | 1041 |
| Surprise | 107 |
| Fear | 40 |
| Disgust | 2 |
| Frustration | 1849 |
| Other | 3 |
| XXX | 2507 |
| **Total** | **10,039** |

Table 3.1: Original IEMOCAP dataset distribution

The sample distribution among the emotion categories has led us to decide **grouping and removing some emotion categories**. Not only the number of samples for some emotions are too small, but also we have considered that some emotions were too similar to be distinguished by a machine learning algorithm. This is the case of **Happiness and Excitement** for which the classification can be confusing even for a human being. We have also decided to group the emotions Sad and Frustrated for the same reasons.

The emotions **Fear, Disgust and Surprise** were too under-represented to be considered part of the dataset, and for this reason they have been removed from the studied dataset. On the other hand, the category **Other** (3 samples) represents other emotions not considered on the list and because of its small volume it has also been removed. The category **XXX** (2507 samples) corresponds to a class representing ambiguous emotions according to the human annotators. Since ambiguity would only add noise into our emotional classifier, we have decided to remove XXX category from our dataset as well.

The resulting dataset has a **total number of 7,376 samples**, and its distribution after the mentioned transformations can be observed in Table 3.2 and Figure 3.2

| **Emotion** | **Number of samples** |
|---|---|
| Anger | 1101 |
| Happiness | 1636 |
| Sadness | 2931 |
| Neutral | 1708 |
| **Total** | **7,376** |

Table 3.2: Final dataset distribution

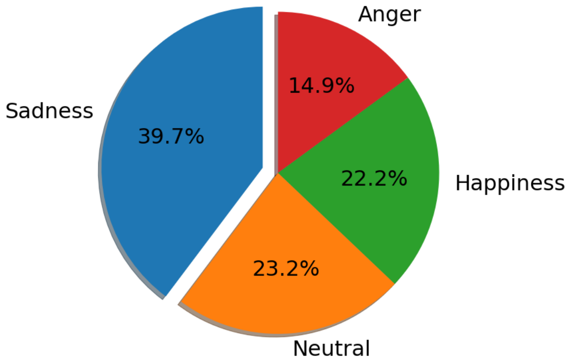


Figure 3.2: Final dataset distribution

## 3.3 Feature Extraction

The following section is focused on explaining the feature extraction methods that we have applied during this work. One of the main goals of this Thesis is to extract and understand which features are most relevant when working with audio and text information together, as well as identifying the possible interference between features. In this section, all the methods and extracted features used for speech and text are explained and will be later analyzed and compared in Chapter 4.

### 3.3.1 Audio Features

The dataset provides the audio files in a WAV format. The first step for extracting features from them is converting those files into audio vectors. To do so, we first read the audio files using the Python library **Librosa** and get the original audio vectors which basically are an **array of amplitudes**.
As each WAV file contains several utterances, we need to truncate the signal at the beginning and end of each utterance. To do so, we take the start and end times associated to each utterance and multiply them by a **sampling rate set to 16000 Hz** to get the actual audio windows for each utterance. Finally, we truncate the original audio vectors with the calculated window to get a more precise bounded audio vector that contains the exact annotated utterance.
(See Figure 3.3)

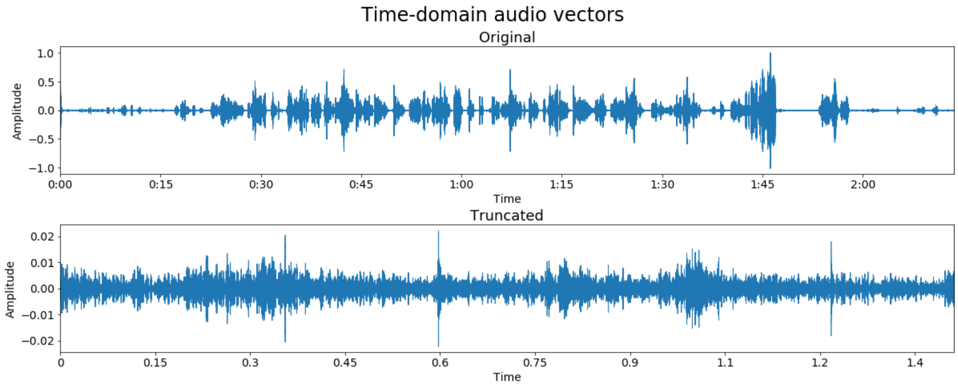


Figure 3.3: Time-domain representation of the original and truncated audio vectors

Once we have each utterance in a sampled audio vector shape, it's time to start extracting the audio features. We have produced two main types of features: **frequency based** features and features representing **prosody attributes** such as pitch an pause.

#### 3.3.1.1 Spectrogram Based Features

Audio signals are usually much more examined in the frequency domain rather than the time domain because usually audio signals change with respect to frequencies, and not time. The frequency domain also shows more information about an audio signal since it shows the audio signal's amplitude at each specific frequency of a bandwidth rather than just the averaged amplitude value.

The main way to represent an audio signal in a frequency domain is by calculating the Fourier Transform of the time signal, which gets a signal in the time domain as input, and outputs its decomposition into frequencies.

In addition, the Short-Time Fourier Transform (STFT), is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment.

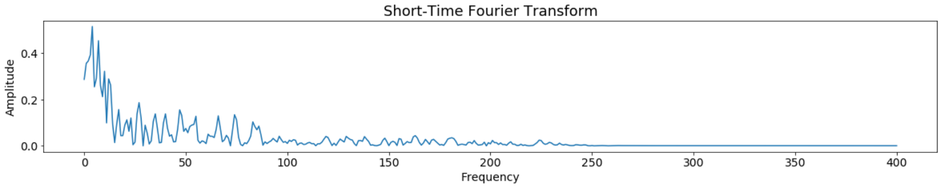


Figure 3.4: Frequency-domain representation of the truncated audio vector

In our case, we have applied a **window size of 800 samples**, which corresponds to a physical duration of **50 milliseconds** at a sample rate of 16,000 Hz. **For each frame, a short term Fourier transform (STFT) of length 800 with hop length 400 is calculated**.

#### 3.3.1.1.1 Mel Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. To calculate the spectrogram for our signals we have taken the complete utterance, separated it into frames, and applied the Fourier Transform on each frame.

Usually, when visualizing a spectrogram, both amplitude and frequency are converted to log scale (dB) to be better represented.

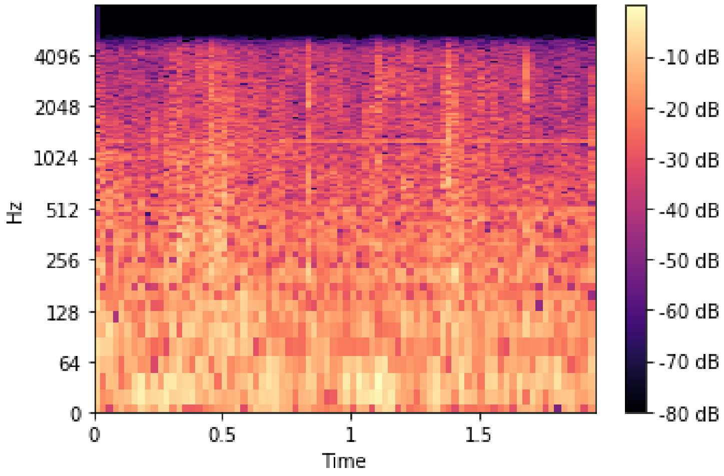


Figure 3.5: Spectrogram in Logarithmic scale

However, studies have shown that humans do not perceive frequencies on a linear scale. We are better at detecting differences in lower frequencies than higher frequencies. For example, we can easily tell the difference between 500 and 1000 Hz, but we will hardly be able to tell a difference between 10,000 and 10,500 Hz, even though the distance between the two pairs are the same.

In 1937, [Volkmann et al., 1937] proposed a unit of pitch such that equal distances in pitch sounded equally distant to the listener. This is called the **mel scale**. To convert frequencies to mel scale we perform a mathematical operation on frequencies. The reference point between this scale and normal frequency measurement is defined by assigning a perceptual pitch of 1000 mels to a 1000 Hz tone, 40 dB above the listener's threshold. Above about 500 Hz, increasingly large intervals are judged by listeners to produce equal pitch increments. As a result, four octaves on the hertz scale above 500 Hz are judged to comprise about two octaves on the mel scale.

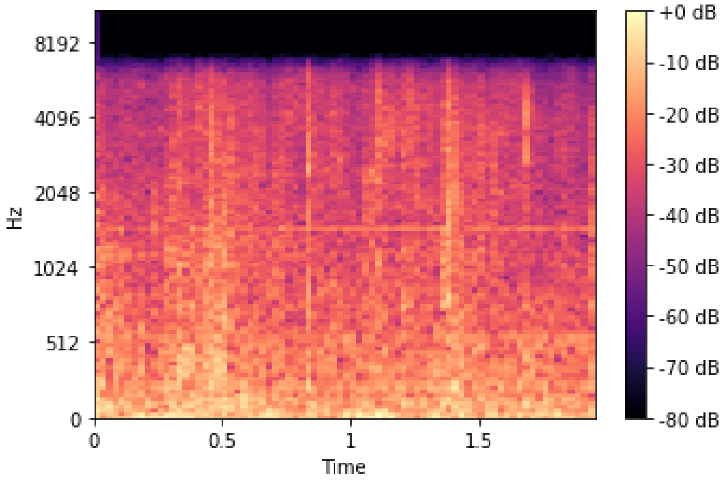


Figure 3.6: Spectrogram in Mel scale

Basically, by converting our signal to Mel scale we get to **mimic the non-linear human ear perception of sound**. To convert our spectrogram in mel scale we have taken the entire frequency spectrum, and separated it into 128 (n mels) frequencies evenly spaced for the human ear.

The Mel-spectrograms are extracted from all utterances using the Python library Librosa, and have served as the basic features for our experiments.

#### 3.3.1.2 Prosodic Features

Emotional prosody is defined as the ability to express emotions through variations of different parameters of the human speech, such as pitch contour, intensity and duration. This ability is probably one of the most basic features of language. However, the study of prosody is very complex because at the same time is both universal across human languages and specific to each one.

Prosody allows communication of both linguistic and emotional intentions at the same time and carries important information related to the sex, age and emotional state of the speaker. Research into the encoding of emotional states in speech signals show clear correlation with global properties, such as loudness, speech rate and pitch contour. For example, depressed and schizophrenic patients [Alpert et al., 2001] typically show reduced emotional prosody expression, and depressed children (9–11 years old) [Emerson et al., 1999] show less ability than non-depressed children to accurately identify affective prosody.

Recently, efforts have been made to explore the specific acoustic features of emotional speech using objective acoustic measures, such as the harmonics, the speech energy and pauses [Besson et al., 2002]. In this work we have experimented with the following handcrafted prosodic features to analyze their influence when classifying an emotion in speech:

- Pitch
- Harmonics
- Speech Energy
- Pause

#### 3.3.1.2.1 Pitch

Pitch is the relative highness or lowness of a tone as perceived by the ear, which depends on the number of vibrations per second produced by the vocal cords. Pitch is the main acoustic correlate of tone and intonation.

We have considered pitch as a relevant feature to be studied, since wave-forms produced by our vocal cords change depending on our emotions. Many algorithms for estimating the pitch signal exists and we have used the most common method, based on auto correlation of center-clipped frames. This method is based on detecting the highest value of the auto correlation function in the region of interest. For given input signal $x(n)$, it gives a resultant clipped signal, $y(n)$:

$$y(n) = \begin{cases} x(n) - C_l, & \text{if } y(n) \geq C_l \\ 0, & \text{if } |y(n)| < C_l \\ x(n) + C_l, & \text{if } y(n) \leq C_l \end{cases} \tag{3.1}$$

where $C_l$ is typically half the mean of the input signal. Later, auto correlation is calculated for the obtained signal $y(n)$ , which is further normalized and the peak values associated with the pitch of the given input $x(n)$.

#### 3.3.1.2.2 Harmonics

The harmonic structure of a vocal sound depends on the wave form produced by the vibrating vocal cords. Like any musical instrument, the human voice is not a pure tone (as produced by a tuning fork); rather, it is composed of a fundamental tone (or frequency of vibration) and a series of higher frequencies called upper harmonics, usually corresponding to a simple mathematical ratio of harmonics. Thus, if a vocal fundamental has a frequency of 100 cycles per second, the second harmonic will be at 200, the third at 300, and so on.

As long as the harmonics are precise multiples of the fundamental, the voice will sound **clear and pleasant**. If non-harmonic components are added (giving an irregular ratio), increasing degrees of **roughness, harshness, or hoarseness** will be perceived in relation to the intensity of the noise components in the frequency spectrum. For this reason we have decided to include harmonics as part of the features to be studied.

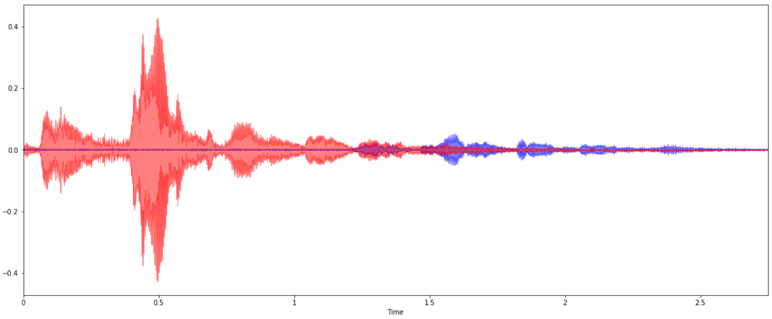


Figure 3.7: Harmonics of angry (red) and sad (blue) audio signals

As it can be observed in Figure 3.8, in the emotional state of anger or for stressed speech, there is an apparent apparent excitation in the spectrum as harmonics and cross-harmonics.

#### 3.3.1.2.3 Speech Energy

The distribution of spectral energy of a speech utterance depends directly on its emotional content. It is observed that high-arousal emotions like happiness or anger have high energies at higher frequencies, while utterances with low-arousal emotions like sadness have less energy in the similar range [Bandela and Kumar, 2017].

To represent speech energy we use Root Mean Square Energy (RMSE) using the equation:

$$E = \sqrt{\frac{1}{n}\sum_{i=1}^{n} y[i]^2} \tag{3.2}$$

RMSE is calculated frame by frame and we take the average and standard deviation as features.

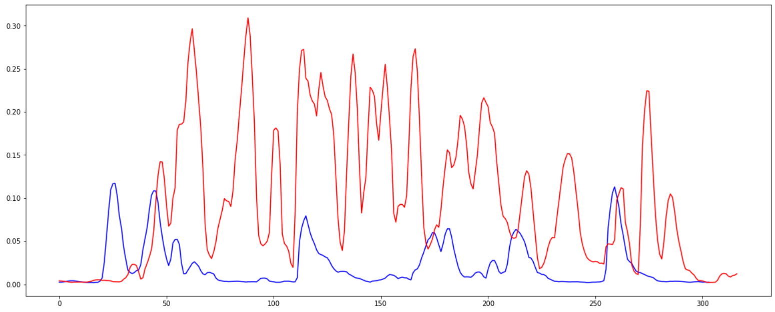


Figure 3.8: RMSE plots of angry (red) and sad (blue) audio signals

#### 3.3.1.2.4 Pause

The relationship between speech pauses and emotions is the research topic of a wide variety of studies [Tisljár-Szabó and Pléh, 2014]. The early studies that investigated the relationship between speech disfluencies (or speech errors) and anxiety, found that in an anxious state, the number and length of pauses increase. Also, recognition of happy and sad sentences is markedly affected by a high speech rate. In most of the sentences, independent of the emotional category, if the utterance is slowed down, the sad emotional content increases, while the frequency of frightened, angry tend to be much faster.

The pause feature aims to represent the silent portion in the audio signal and is extracted using the RMSE with the following equation:

$$Pause = Pr(y[n] < t) \tag{3.3}$$

where $t$ represents a chosen threshold $\approx 0.4 * RMSE$

### 3.3.1.3 Audio Feature Extraction Pipeline

The pipeline followes to extract the features from audio files is summariez in Figure 3.9

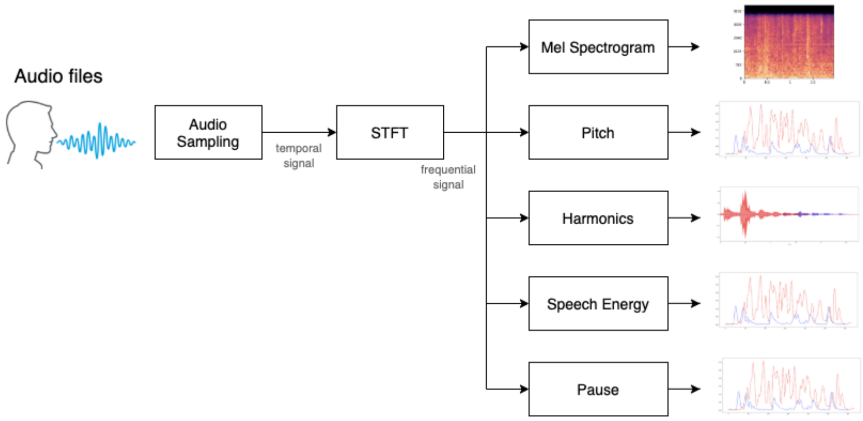


Figure 3.9: Audio Feature Extraction Pipeline

### 3.3.2 Text Features

The textual information for each speech utterance is provided by IEMOCAP in a short sentence shape. In order to extract relevant features, we first apply several text processing steps.

#### 3.3.2.1 Text Normalization

In the first place, non alphabetical characters are removed from each transcription and converted to **lower case characters**. Symbols like commas and hyphens are removed from the transcripts and sentences are converted into plain text. After some experiments, we have decided to **keep question marks and exclamation points** since they seem to add emotional value to expressions in terms of **arousal**, for emotions like Anger, Surprise or Joy (e.g. ”are you talking to me!?” and ”are you talking to me”).

#### 3.3.2.2 Stop-words

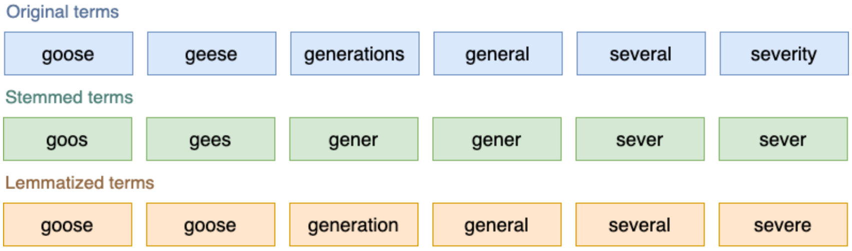
One of the major forms of text pre-processing is to filter out useless data. In natural language processing, useless words are referred to as stop words. A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) which do not contain important significance to be used in search queries. Usually, these words are filtered out from search queries because they return a vast amount of unnecessary information.

Using a stop list significantly reduces the number of postings that a system has to store, but in our case we have considered that words considered as **stop words do have significant emotional content** (e.g. What!?) and for this reason we have decided to keep all the words contained in the stop list. Since utterances are usually short and do not contain a lot of words, we decided to avoid removing the less content possible from each transcription.

#### 3.3.2.3 Stemming and Lemmatization

Stemming is the process of reducing inflection in words to their root forms, such as mapping a group of words to the same stem, even if the stem itself is not a valid word in a language. On the other hand, Lemmatization, unlike stemming, reduces the inflected words properly ensuring that the root word belongs to the language. In Lemmatization the root word is called a Lemma, which is the canonical form, dictionary form, or citation form of a set of words.

As an experiment, we tried building a text corpus by stemming the words in each transcript but found out that this technique reduced too much the meaning contained in the transcribed words. On the contrary, lemmatization allowed us to convert words into their most standard forms, removing the less meaning possible from them. For this reason we have decided to use lemmatization instead of stemming in our approach.



| | | | | | |
|---|---|---|---|---|---|
| Original terms | | | | | |
| goose | geese | generations | general | several | severity |
| Stemmed terms | | | | | |
| goos | gees | gener | gener | sever | sever |
| Lemmatized terms | | | | | |
| goose | goose | generation | general | several | severe |

Figure 3.10: Stemmed and lemmatized terms

#### 3.3.2.4 Text Features Extraction

Machine learning algorithms cannot work with raw text directly. Rather, the text must be converted into vectors of numbers. The process of converting text into vectors is known as textual feature extraction. In this section we explain a couple of methods for feature extraction and our chosen approach.

##### 3.3.2.4.1 Bag of Words

In natural language processing, a common technique for extracting features from text is to place all of the words that occur in the text in a bucket. This approach is called a **bag of words** model or BoW for short. It's referred to as a "bag" of words because any information about the structure of the sentence is lost. The BoW model is the simplest form of text representation in numbers. Like the term itself, we can represent a sentence as a bag of words vector (i.e. a string of numbers).

The model throws away all of the order information in the words and focuses on the occurrence of words in a document. This can be done by assigning each word a unique number. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector is filled with a count or frequency of each word in the encoded document.

For example, given the following sentences:

- **Sentence 1**: This movie is very scary and long
- **Sentence 2**: This movie is not scary and is slow

would result in a BoW model such as:

| | **this** | **movie** | **is** | **very** | **scary** | **and** | **long** | **not** | **slow** |
|---|---|---|---|---|---|---|---|---|---|
| **Sentence 1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| **Sentence 2** | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 1 | 1 |

Table 3.3: BoW model representation

This is the bag of words model, where we are only concerned with encoding schemes that represent what words are present or the degree to which they are present in encoded documents without any information about order. The BoW model represented in Table 3.3 is built on a vocabulary of 9 different words. If we wanted to extract features from a sentence that contained new words, we would need to rebuild our BoW model.

Additionally, the vectors resulting from a BoW model may contain many 0's, thereby resulting in a sparse matrix and containing few information. Finally, with BoW we are retaining no information on the grammar of the sentences nor on the ordering of the words in the text.

#### 3.3.2.4.2 TF-IDF

One of BoW drawbacks is that all words appearing in the vocabulary have the same importance in the model. A possible strategy to score the relative importance of words is by using the so called method Term Frequency-Inverse Document Frequency (TF-IDF).

The concept of **Term Frequency (TF)** corresponds to number of times a word appears in a document divided by the total number of words in the document:

$$tf_{t,d} = \frac{n_{t,d}}{Number\ of\ terms\ in\ the\ document} \tag{3.4}$$

On the other hand, **Inverse Document Frequency (IDF)** is a measure that represents how important a term is in a document. In other words, while TF represents how common a word is in a document, IDF measures how rare the word is in the same context. IDF can be calculated like this:

$$idf_t = log \frac{number\ of\ documents}{number\ of\ documents\ with\ term\ 't'} \tag{3.5}$$

Now, the **TF-IDF score** can be computed for each word in the corpus as the product of TD and IDF scores:

$$(tf\_idf)_{t,d} = tf_{f,d} * idf_t \tag{3.6}$$

Words with a higher score are more important, rare and therefore more significant than those with a lower score. Given the nature of the corpus formed by IEMOCAP transcriptions, which are usually short, and non specific to any topic, we have considered TF-IDF to be a suitable method for extracting features from text.

#### 3.3.2.5 Text Feature Extraction Pipeline

The pipeline followed to extract features from transcripts is shown in Figure 3.13.

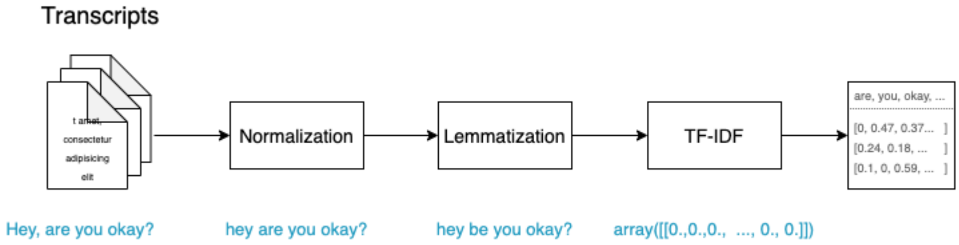


Figure 3.11: Text Feature Extraction Pipeline

## 3.4 Model architectures

In this work we have used two types of architectures. On one hand, we use traditional machine learning models to test the performance of the extracted features. We compare different types of classifier to understand the impact of each type of feature and how they interfere with each other when using each model to classify an emotion.

On the other hand, once we have proved which is the best combination of features, we define a multimodal model using a deep neural architecture for acoustic and textual data. We also compare the classification performance achieved by traditional models with the deep neural architecture results. In this section we first explain the traditional ML classifiers used for the feature validation experiments. Afterwards, we describe the deep neural architectures used in our multimodal model.

## 3.4.1 Traditional models

This section describes the various ML-based classifiers used for feature validation, namely, Random Forests, Gradient Boosting, Support Vector Machines, Naive-Bayes, and Logistic Regression.

### 3.4.1.1 Random Forest

Random forests are ensemble learners that operate by constructing multiple decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. It has two base working principles:

- Each decision tree predicts using a random subset of features
- Each decision tree is trained with only a subset of training samples. This is known as bootstrap aggregating

Finally, a majority vote of all the decision trees is taken to predict the class of a given input.

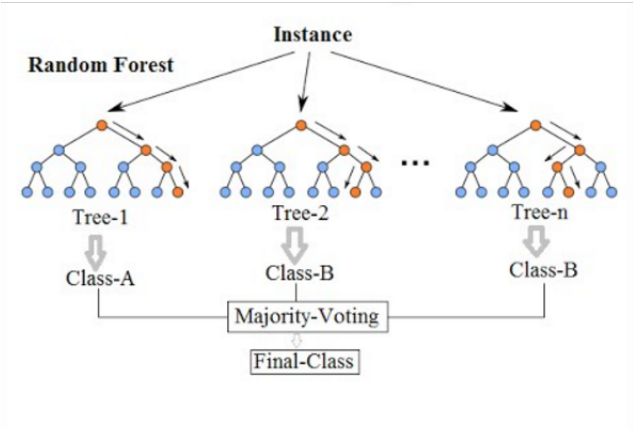


Figure 3.12: Random Forest classification. *Source: medium.com*

### 3.4.1.2 Gradient Boosting (XGB)

XGB refers to eXtreme Gradient Boosting, which is an implementation of boosting that supports training the model in a fast and parallelized way. Boosting is another ensemble classifier combining a number of weak learners, typically decision trees. They are trained in a sequential manner, unlike Random Forests, using forward stagewise additive modeling.

During the early iterations, the decision trees learned are simple. As training progresses, the classifier becomes more powerful because it is made to focus on the instances where the previous learners made errors. At the end of training, the final prediction is a weighted linear combination of the output from the individual learners.

#### 3.4.1.3 Support Vector Machine (SVM)

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. An SVM training algorithm essentially builds a non-probabilistic binary linear classifier.

It represents each training example as a point in space, mapped such that the examples of the separate categories are divided by a clear gap that is as wide as possible (this is usually achieved by minimizing the hinge loss). New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

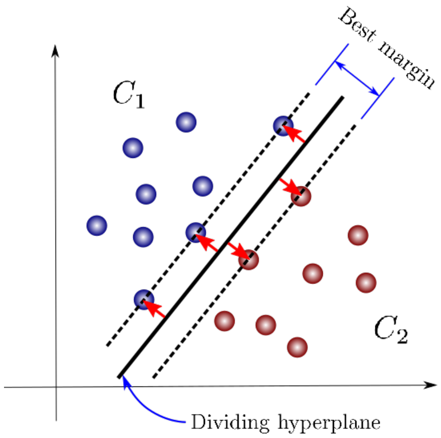


Figure 3.13: Support Vector Machine classification. *Source: towardsdatascience.com*

SVMs were originally introduced to perform linear classification; however, they can efficiently perform a non-linear classification using the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. In this work, we have considered two types of kernel: linear and radial basis function.

#### 3.4.1.4 Gaussian Naive Bayes (GNB)

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong naive independence assumptions between the features.

Under gaussian settings, the feature vectors represent the frequencies with which certain events have been generated by a gaussian $(p_1, ..., p_n)$ where $p_i$ is the probability that event $i$ occurs. GNB is very popular for document classification task in text, which too essentially is a multi-class classification problem.

#### 3.4.1.5 Logistic Regression Classifier (LR)

Logistic Regression classifiers are typically used for binary classification problems, that is, when we have only two labels. In this work, LR is implemented in a one-vs-rest manner; four classifiers (one per emotion) have been trained for each class and finally, we consider the class that is predicted with the highest probability.

#### 3.4.1.6 Ensemble of models

Having trained the above classifiers, we take the ensemble of the best performing classifiers and use it for comparison as a separate classifier. This technique is known as **stacking** and it can be described as an ensemble learning technique where the predictions of multiple classifiers (referred as level-one classifiers) are used as new features to train a meta-classifier.

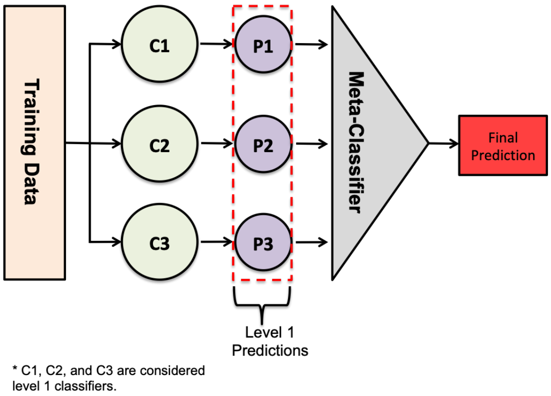


Figure 3.14: Stacking classifier framework. *Source: towardsdatascience.com*

Figure 3.14 shows how three different classifiers get trained. Their predictions get stacked and are used as features to train the meta-classifier which makes the final prediction. To prevent information from leaking into the training from the target, **the level one predictions should come from a subset of the training data that was not used to train the level one classifiers**. A possible way to do this is by using $k$-fold cross validation to generate the level one predictions. First, the training data is split into $k$-folds. Then the first $k-1$ folds are used to train the level one classifiers. The validation fold is then used to generate a subset of the level one predictions. The process is repeated for each unique group.

### 3.4.2 Deep Learning models

In this section we describe the deep neural architectures that we have built and experimented with, using the features described in Sections 3.3.1 and 3.3.2 with the goal to compare their performance with the classical machine learning models presented in the previous section and with the current state-of-the-art for emotion recognition on the IEMOCAP dataset.

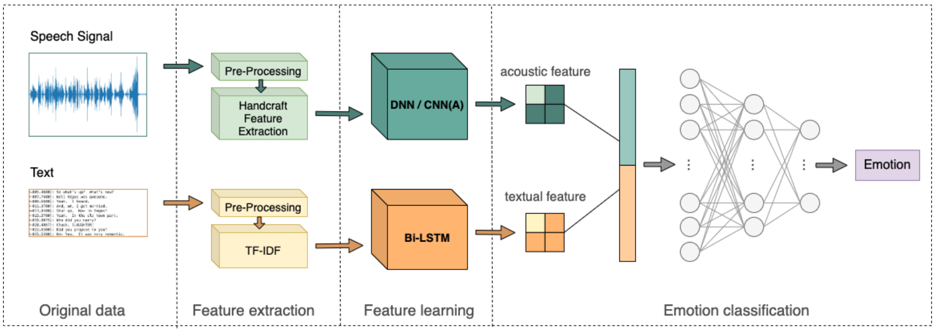


Figure 3.15: The structure of our multimodal model

Figure 3.15 represents the whole flow of our multimodal model. We have tried three different approaches to solve the audio feature learning process, namely a Dense Neural Network (DNN), a Convolutional Neural Network with a a Bi-LSTM (CBL) and the latter with attention (CBLA).
The text features are learned using a bidirectional LSTM (BLSTM). In the following sections, each architecture is further explained.

#### 3.4.2.1 DNN/BLSTM

The first proposed architecture uses a Dense Neural Network for audio feature learning, and a Bi-LSTM for textual feature learning. Then, a feature fusion method is resorted to merge the emotional features of audio and text, and additional dense layers are used to classify the fusion features. Figure 3.17 represents the architecture of the network for this first approach.

#### 3.4.2.1.1 BLSTM for textual feature learning

Long Short Term Memory networks (LSTM) are a special kind of Recurrent Neural Network (RNN), **capable of learning long-term dependencies**. LSTM relies on its three gates structure, which effectively solves the long-term dependence in the neural network, and avoids the gradient disappearance problem in the common recurrent neural network, and it is suitable for the modelling of speech temporal signals and text signals which are closely related to time.

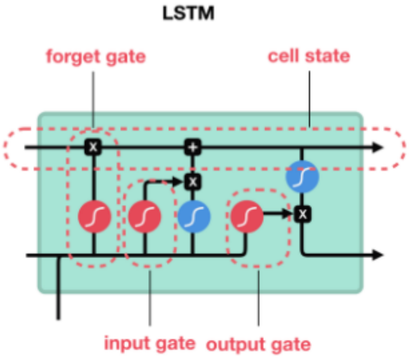


Figure 3.16: Structure of a LSTM cell

LSTMs, like all RNNs, have the form of a chain of repeating modules (cells) of neural network. The first step in a LSTM consists in determining what information to lose from the cell through the forgetting gate. The next step is deciding how much new information is added to the cell state. The output gate then uses a sigmoid layer to determine which cell states to output.

The bidirectional LSTM consists of two ordinary LSTM, a forward one which uses the past information and an inverse one that obtains the future information. In this way, the information at $t-1$ as well as at $t+1$ all can be used at time $t$. It would be more accurate than LSTM and can avoid the long-term dependence problem in features learning. Hence, it can be utilized for the textual emotion feature extraction. We feed the textual TF-IDF vectors into the bidirectional long-short-term- memory network to extract high-level information.

#### 3.4.2.1.2 DNN for audio feature learning

A Dense Neural Network (DNN) is a network which layers are fully connected (dense) by the neurons in a network layer. Each neuron in a layer receives an input from all the neurons present in the previous layer. A **densely connected layer provides learning features from all the combinations of the features** of the previous layer, whereas a convolutional layer relies on consistent features with a small repetitive field. Hence, we have used a DNN for learning audio features.

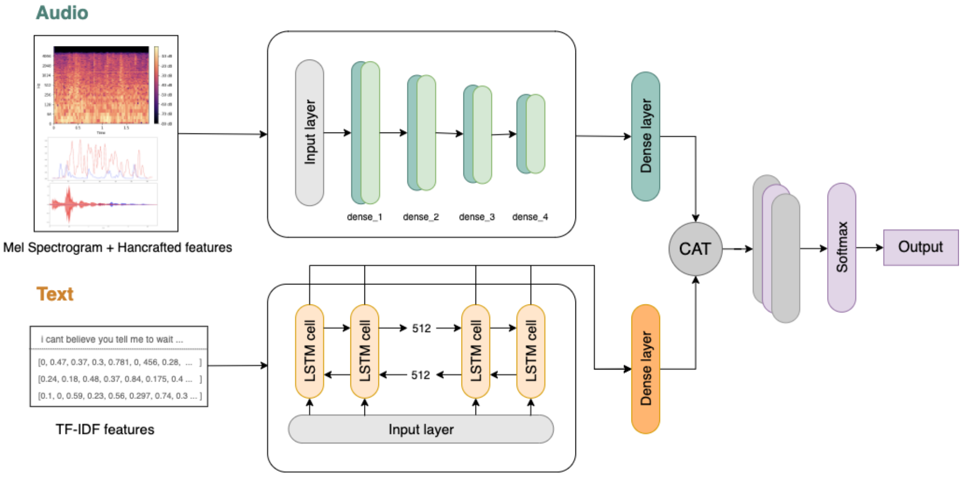


Figure 3.17: DNN/BLSTM network architecture

#### 3.4.2.1.3 Feature fusion

After feeding acoustic features into the DNN and textual into the Bi-LSTM network, we obtain the high-level textual features and acoustic features which consist of global and local information. In this work, we adopt the feature-level fusion approach. The advantage is that emotional features extracted from different modes are directly related to the final decision, and the fusion results can retain the feature information needed in the final decision to the greatest extent.

The final descriptor of the multimodal emotion features vector is created using an ordered concatenation of textual and acoustics features. After that, we feed the fusion emotion feature vector into a deep neural network containing four dense layers and a softmax layer to capture the associations between the features from different modalities. **The output of softmax represents the relative probability between different emotion classes**

$$p(x_a) = softmax(x_a) = \frac{exp(x_a)}{\sum_{a'}^{A} exp(x_{a'})} \tag{3.7}$$

where $a$ represents the emotion categories and $p(x_a)$ represents the probability of $a$-th category.

In addition, to avoid overfitting, we added regularization in the multimodal features training. The principle of regularization is to add an index to describe the complexity of the model in the loss function.

### 3.4.2.2 CBL/BLSTM

The second approach we have adopted has been a Convolutional Neural Network and a Bi-LSTM (CBL) for audio feature learning, and keeping the Bi-LSTM approach for text feature learning. Figure 3.18 represents this approach architecture.

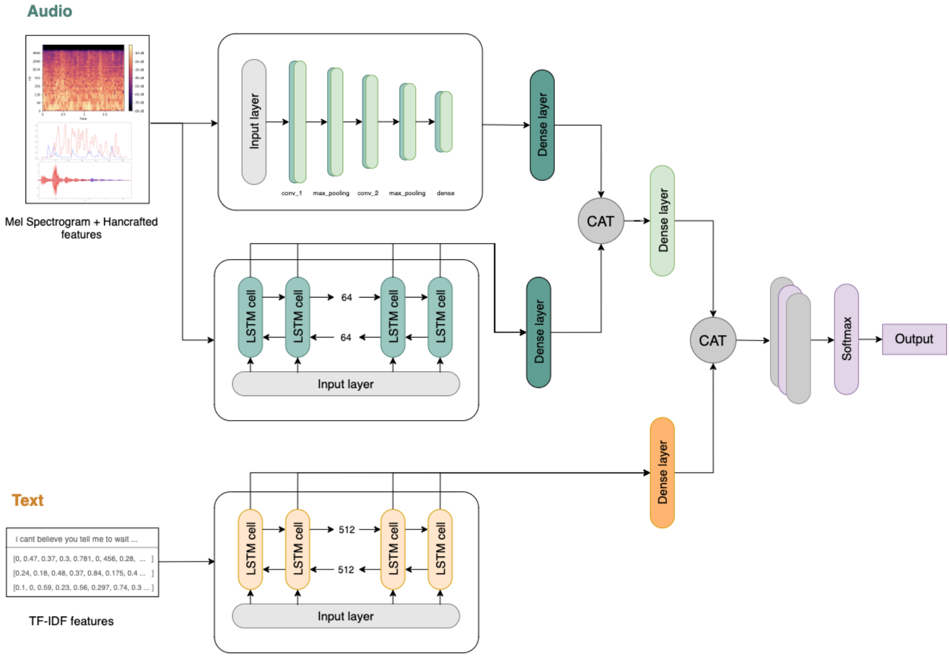


Figure 3.18: CBL/BLSTM network architecture

Convolutional neural network (CNN) is one of the common deep learning neural networks. The infrastructure of CNN includes the convolution layer, pooling layer and dense layer. CNN can extract some advanced characteristics automatically. Convolution layer weight sharing reduces the complexity of the network model, alleviates overfitting, pooling operation reduces the number of neurons, and is more robust to the translation of input space.

CNN is a process from local to global (local to global realization is in the dense layer) while the traditional neural network is the entire process. CNN network can reduce variance in frequency of the input and captures local information, but without considering the global features and context.

Voice is kind of a nonlinear time series signal; text information is closely related to temporal context, and they are all time-related. Therefore, it is the LSTM network which is suitable for acoustic and text feature extraction and learning that models in context and helps to learn the relevance of features. But in LSTM, there is no intermediate nonlinear hidden layer that causes the increase in variation in the hidden state factors. In brief, the model capabilities of CNN and LSTM are both limited.

#### 3.4.2.3 CBLA/BLSTM

In the CBL network, the output of a set of CNN networks was thrown in LSTMs directly. From this way, we can get the high-level information which contains both local information and long-term contextual dependencies. However, CNNs focus on local information and discard a lot of data.

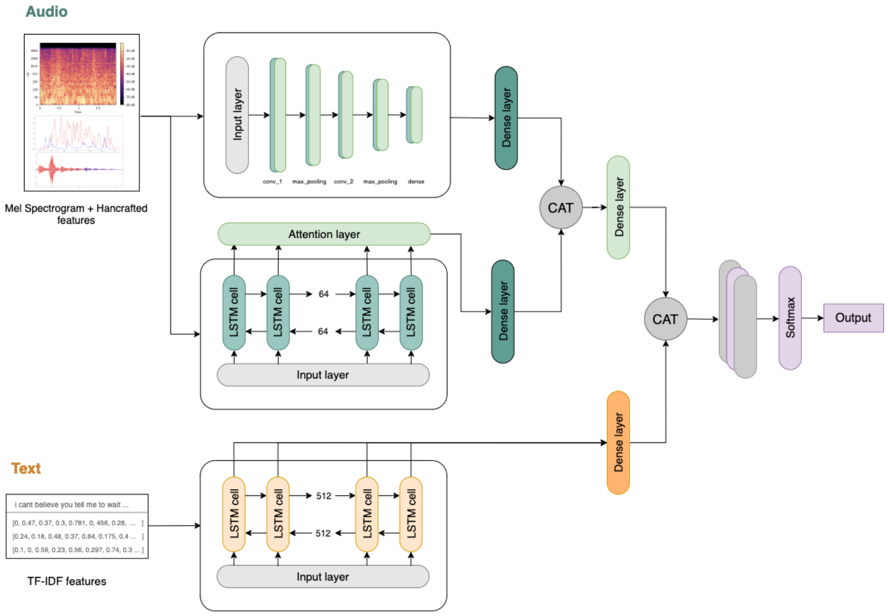


Figure 3.19: CBLA/BLSTM network architecture

To avoid valuable data losing, we constructed the model CBLA which uses binary channels of CNN and Bi-LSTM. In the CNN channel, we constructed four one-dimensional convolution layers with different filters' number and cropped for one-dimensional temporal input (audio features).

We employed the maximum pooling layer and global average pooling layer to carry out the maximum pooling operation and global average pooling operation for the data. In the Bi-LSTM channel, a set of BiLSTM cells were put up to extract longterm contextual dependencies information, and an attention mechanism was added to find more effective features.

At last, the data from two channels were concatenated and the output was put into a three dense layers. After the nonlinear change of the dense layer, the correlation between these features was extracted and finally mapped to the output space. The structure of the CBLA model is shown in Figure 3.19

This approach is the same as the one in Section **??** with adding an attention layer. A neural attention mechanism equips a neural network with the ability to focus on a subset of its inputs (or features). This method is most commonly used in sequence-to-sequence models to attend to encoder states, but can also be used in any sequence model to look back at past states. Using attention, we obtain a context vector $C_i$ based on hidden states $s_1, ..., s_m$ that can be used together with the current hidden state $h_i$ for prediction. The context vector $c_i$ at position is calculated as an average of the previous states weighted with the attention scores $a_i$:

$$c_i = \sum_j a_{ij} s_j \tag{3.8}$$

$$a_i = softmax(f_{att}(h_i, s_j)) \tag{3.9}$$

In our approach, we use an additive type of attention. This is the original attention mechanism [Bahdanau et al., 2015], which uses a one-hidden layer feed-forward network to calculate the attention alignment:

$$f_{att}(h_i, s_j) = v_a^\top tanh(W_a[h_i; s_j]) \tag{3.10}$$

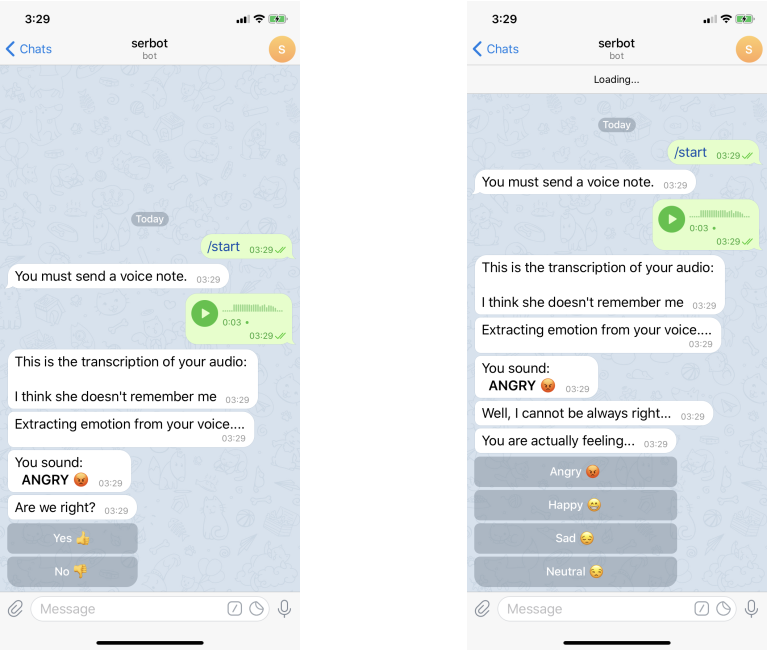
where $v_a$ and $W_a$ are learned attention parameters.

## 3.5 Emotion Classification

The final step in our pipeline defined for MSER is the emotion classification per se. The goal of SER consists in recognizing emotions from unseen audio samples, and to do so we need to make predictions upon a trained model.

In order to test our models in a non-theorical environment, we have built a mechanism for model consumption and prediction for an unseen audio sample. The idea behind this is to assess the performance of the system in front of real life audio samples, considering environmental noises and variations in tones which aren't provided by IEMOCAP. However, this application also aims to equip the system with a continuous retraining and self-improvement mechanism by receiving feedback for misclassified samples.

To accomplish this, we have built a Telegram bot using the Python Telegram API **telepot**. The user must send a recorded voice message and the bot makes an emotion classification using the best performing model in production. The predicted emotion is displayed for the user to give his feedback. Once the user has determined whether it's been a correct or incorrect classification and indicated the right emotion, the audio sample and the correct emotion are stored to be later used for a model retraining.



/start 03:29

You must send a voice note. 03:29

0:03 03:29

This is the transcription of your audio:

I think she doesn't remember me 03:29

Extracting emotion from your voice.... 03:29

You sound:
ANGRY 😡 03:29

Are we right? 03:29

/start 03:29

You must send a voice note. 03:29

0:03 03:29

This is the transcription of your audio:

I think she doesn't remember me 03:29

Extracting emotion from your voice.... 03:29

You sound:
ANGRY 😡 03:29

Well, I cannot be always right... 03:29

You are actually feeling... 03:29

Figure 3.20: Bot feedback retrieval

# Chapter 4

# Experiments

Having defined a series of light machine learning classifiers and three deep neural approaches, this section is devoted to explain the experiments performed with each of them with the extracted acoustic and textual features.

The experiments were designed with three main goals in mind:

- Prove the validity of the extracted features for both audio and text
- Compare the light classifiers with the proposed deep neural architectures in terms of performance
- Testing the overall performance of the system over an unseen recorded audio sample.

## 4.1 Combination of Features

For the sake of proving feature validity and testing the improvement in accuracy that each feature supposed to each other, we have separated audio features by type and combined them with text features.

1. **Mel audio features**: Mel Spectrogram features
2. **HC audio features** (Handcrafted): Pitch, Pause, Speech energy and Harmonics features
3. **TF-IDF text features**: TF-IDF features

Each type of features has been tested individually and combined with each other. Table 4.1 illustrates the seven feature combinations we have used in our experiments.

| | Audio | | Text |
|---|---|---|---|
| | **Mel** | **HC** | **TF-IDF** |
| **Audio Basic** | X | | |
| **Audio HC** | | X | |
| **Audio Extra** | X | X | |
| **Combined Basic** | X | | X |
| **Combined HC** | | X | X |
| **Combined Extra** | X | X | X |
| **Text Basic** | | | X |

Table 4.1: Feature combinations

## 4.2 Experiments with traditional models

In this section we explain the different settings in which we conducted our experiments with the traditional Machine Learning models, namely Random Forest, XGBoost, Linear Regression classifier, Support Vector Machine, Multi-Layer Perceptron and Gaussian Naive Bayes classifier.

- We use Librosa Python library to process the audio files and extract the features described in Section 3.3.1.
- We use scikit-learn and xgboost (machine learning libraries for Python) to implement all the ML classifiers

As the dataset is not explicitly split beforehand into training and testing sets, we perform 5-fold cross validation to determine the overall performance of the model. The data in each fold are split into training and testing datasets (8:2, respectively). The hyper-parameters for all classifiers have been chosen by using a Grid Search to find the most optimal estimator. The estimators that have admitted it, have been trained using L2 regularization to avoid overfitting. In the cases where we have detected a bias problem, we have adjusted our models to decrease the regularization parameters.

The ensemble model resulting from stacking the best performing models has been implemented using a Scikit-learn stacking classifier (as described in section 3.4.1.6).

The results achieved by all ML models have been compared with the performance of a "Dummy" model, which always classifies an emotion as the majority class. This is done in order to assess the contribution of each model in terms of relative quality.

## 4.3 Experiments with Deep Neural architectures

The experiments conducted on the three proposed architectures in Section 3.4.2 use the multimodal combinations of features described in Table 4.1. That is, Combined Basic (Text + Mel features), Combined HC (Text + Handcrafted features) and Combined Extra (Text + Mel + Handcrafted features).

The three architectures (DNN/BLSTM, CBL/BLSTM, CBLA/BLSTM) are implemented using keras and optimized using the Adam method. We have trained each model on 100 epochs with a batch size of 256 samples and saved the best one as the final model.

We use a learning rate decay, which reduces learning rate when a metric (in our case, validation loss) has stopped improving. We also set an early stopping parameter with a patience equivalent to 5, that stops our training if it hasn't improved for the last five epochs. We use categorical cross entropy as loss function, and accuracy as the performance metric.

# Chapter 5

# Results

In this section we present the results achieved in all the conducted experiments. By comparing the performance of every setting, we aim to validate the fact that using text and audio features together, results in an improvement over emotion classification.

We also draw from the premise that there must be a performance improvement when using handcrafted features over basic features, and that the proposed deep neural approaches are expected to provide better results than classical machine learning classifiers.

The results are presented in two sections. First, we compare the performance of traditional ML classifiers on each combination of features and analyze the best performing setting. Afterwards, we test each deep neural approach with three combinations of multimodal features.

## 5.1 Traditional model results

The models compared in these experiments have been Random Forest, XGBoost, Gaussian Naive Bayes, Logistic Regression, Support Vector Classifier and Multilayer Perceptron.

In order to compare all models between each other, we have first extracted a table of metrics and also represented their mean accuracy and standard deviation on boxplot figures. Then, for some models we also show the learning curves for training and validation scores to try understanding which approach could improve their performances.

### 5.1.1 Audio Basic Features

This experiment is conducted over Mel Spectrogram audio features only. Six ML classifiers have been trained and their performance compared (see Figure 5.1). While GNB performs specially poorly with Mel features, Random Forest and XGBoost achieve 0.474 and 0.493 of accuracy respectively. The ensemble classifier stacks these two models but doesn't overcome their results.

| **model** | **accuracy** | **f1-score** | **precision** | **recall** |
|---|---|---|---|---|
| Dummy | 0,400 | 0,143 | 0,100 | 0,250 |
| LR | 0,441 | 0,307 | 0,465 | 0,333 |
| MLP | 0,470 | 0,402 | 0,465 | 0,399 |
| SVC | 0,444 | 0,303 | 0,404 | 0,345 |
| RF | 0,474 | 0,412 | 0,468 | 0,405 |
| XGB | **0,493** | **0,441** | **0,490** | **0,432** |
| GNB | 0,415 | 0,197 | 0,443 | 0,278 |
| ENSEMBLE | 0,490 | 0,439 | 0,488 | 0,428 |

Table 5.1: Performance metrics for Basic Audio features

Naive Bayes is so called because the independence assumptions that it makes are indeed very naive for a model of natural language. The conditional independence assumption states that features are independent of each other given the class. This is hardly ever true for terms in documents, and is also the case for mel spectrogram features.

Also, Naive Bayes works best when having small training data set, and relatively small features (dimensions). In the case of mel spectrogram, where we have a big set of features (384 columns), the model may not give accuracy because the likelihood is most probably distributed and may not follow the Gaussian or other distribution.

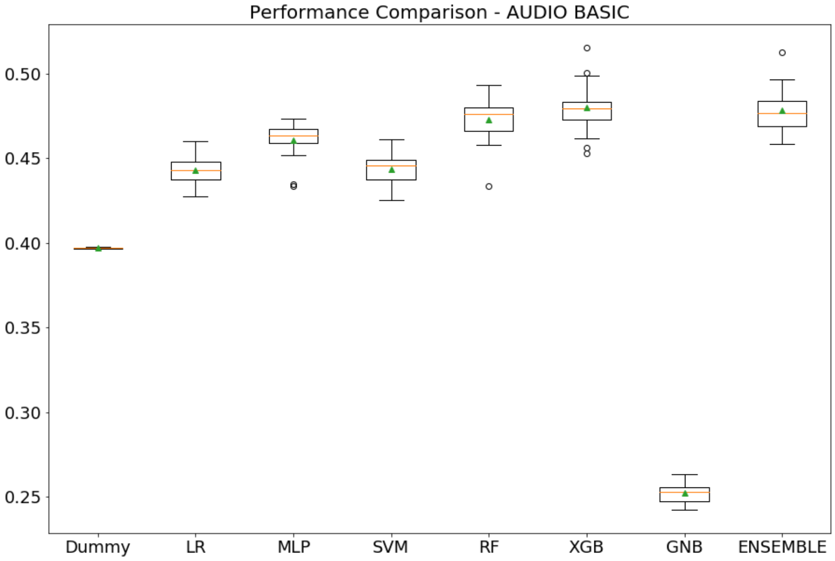


Figure 5.1: ML models performance over Audio Basic features

### 5.1.2 Audio Handcrafted Features

This experimented was made using only handcrafted features. Overall, the performance is worse than when using mel spectrogram features. In this case, the ensemble model combines RF and XGB but doesn't overcome their results either.

| **model** | **accuracy** | **f1-score** | **precision** | **recall** |
|---|---|---|---|---|
| Dummy | 0,379 | 0,137 | 0,095 | 0,250 |
| LR | 0,430 | 0,295 | 0,338 | 0,350 |
| MLP | 0,432 | 0,298 | 0,339 | 0,355 |
| SVC | 0,419 | 0,250 | 0,350 | 0,322 |
| RF | **0,455** | **0,401** | 0,448 | **0,404** |
| XGB | 0,448 | 0,377 | **0,456** | 0,386 |
| GNB | 0,412 | 0,246 | 0,336 | 0,310 |
| ENSEMBLE | 0,448 | 0,365 | 0,455 | 0,381 |

Table 5.2: Performance metrics for Audio Handcrafted features

On the other hand, the only model that performs slightly better in this case is the Gaussian Naive Bayes classifier. This might mean that pitch, pause, speech energy and harmonics are less correlated between each other than in the case of mel spectrogram. However, the overall accuracy when using handcrafted features only is quite poor for all classifiers.

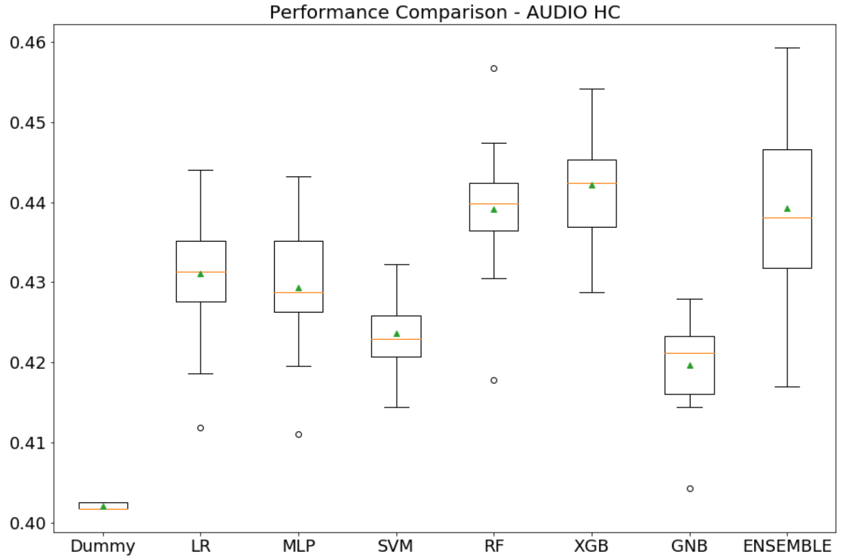


Figure 5.2: ML models performance over Audio Handcrafted features

In order to better understand the value added by handcrafted features, we have printed the learning curves for the XGBoost Classifier. A learning curve shows the validation and training score of an estimator for varying numbers of training samples. It is a tool to find out how much we benefit from adding more training data and whether the estimator suffers more from a variance error or a bias error.

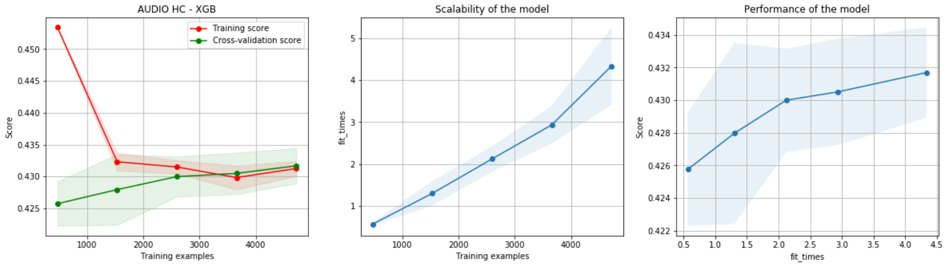


Figure 5.3: Learning curves of XGB over acoustic handcrafted features

From Figure 5.3 we observe that validation and accuracies scores converge at a low level accuracy. This curves indicate a high bias in the model, meaning that we wouldn't benefit from adding more training data.

### 5.1.3 Audio Extra Features

The audio extra features combine Mel spectrogram with handcrafted features. The performance of ML classifiers seem to improve slightly when using the two types of features together. The ensemble classifier in this case also combines RF and XGBoost, but this time it improves their respective metrics.

| model | accuracy | f1-score | precision | recall |
|---|---|---|---|---|
| Dummy | 0,393 | 0,141 | 0,098 | 0,250 |
| LR | 0,448 | 0,324 | 0,462 | 0,353 |
| MLP | 0,480 | 0,435 | 0,484 | 0,424 |
| SVC | 0,450 | 0,321 | 0,391 | 0,359 |
| RF | 0,493 | 0,431 | 0,502 | 0,422 |
| XGB | 0,491 | 0,442 | 0,494 | 0,432 |
| GNB | 0,255 | 0,175 | 0,315 | 0,280 |
| ENSEMBLE | **0,497** | **0,451** | **0,506** | **0,437** |

Table 5.3: Performance metrics for Audio Extra features

In this case, the ensemble model is the best classifier, performing slightly better than XGBoost. Gaussian Naive Bayes performs very poorly due to the presence of mel spectrogram features, emphasizing the difference between all other classifiers.

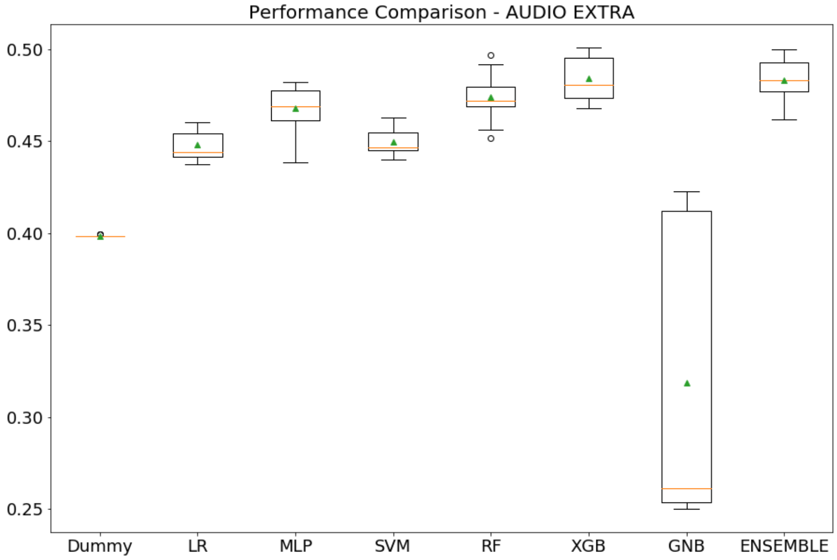


Figure 5.4: ML models performance over Audio Extra features

By printing the learning curves for the Random Forest Classifier, we have realized that training and cross-validation scores don't get to converge at any point and the training score is much higher than the validation score (see Figure 5.6). Thus, by increasing the size of the training set we would benefit from a higher validation score. Also, the performance of the model keeps increasing for each fit. For this reason we conclude that this experiment would perform better for all classifiers with more available training data.

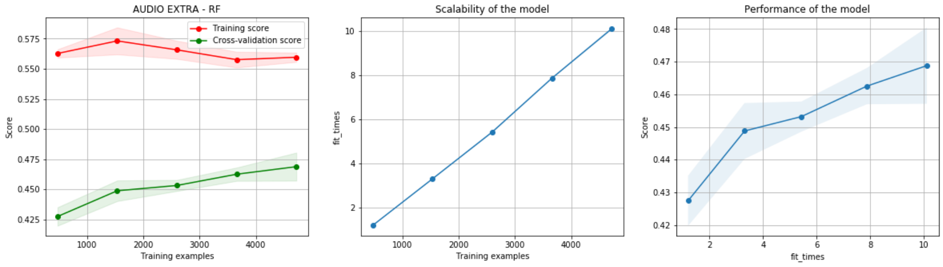


Figure 5.5: Learning curves of Random Forest model over audio extra features

This experiment proves that using handcrafted features improves the performance of classical acoustic features. However, it also demonstrates that in order to **classify emotions with only audio features and higher accuracy, we need much more data for training**.

### 5.1.4 Text Features only

In connection to the experiments conducted using just audio features, the results for this experiment have worked much better. While using Spectrogram and handcrafted features we get a maximum accuracy of 0.497, textual TF-IDF features provide us with a maximum accuracy of 0.602 (see Table 5.4). In this case, the Ensemble model stacks Support Vector and Logistic Regression classifiers, which are the two best performing models.

| model | accuracy | f1-score | precision | recall |
|---|---|---|---|---|
| Dummy | 0,400 | 0,143 | 0,100 | 0,250 |
| LR | 0,587 | 0,567 | 0,597 | 0,553 |
| MLP | 0,562 | 0,549 | 0,560 | 0,541 |
| SVC | **0,602** | 0,566 | **0,627** | 0,547 |
| RF | 0,538 | 0,465 | 0,626 | 0,449 |
| XGB | 0,539 | 0,495 | 0,565 | 0,477 |
| GNB | 0,545 | 0,550 | 0,565 | **0,565** |
| ENSEMBLE | 0,598 | **0,571** | 0,612 | 0,558 |

Table 5.4: Performance metrics for Text features

Overall, all classifiers highly improve the behavior of a dummy classifier, but still they are far from state-of-the-art results.

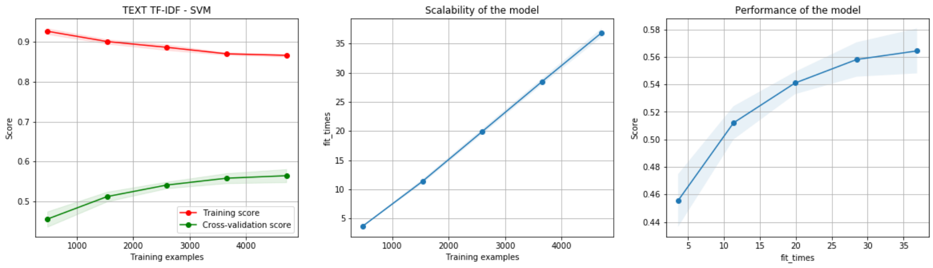


Figure 5.6: Learning curves of Support Vector Classfier (SVC) over text features

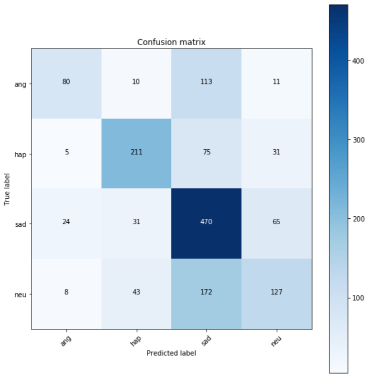


Figure 5.7: SVC Confusion matrix with text features

The learning curves for SVC present a low bias and a high variance. The training score is high (even well higher than the validation score), but the gap between the two scores is big. The model seems to be suffering from overfitting and at this point it may be beneficial to increase the size of the dataset. It exists the possibility for the validation score curve to continue to increasing and converge with the training score curve (which still has room to grow with respect to the desired performance).
Using only textual features results in a relatively important confusion between angry and sad classes. However, the biggest misclassification is between neutral and sad classes.

Still, even if textual features contributions result on better performances comparing to acoustic features, all classifier accuracies are still far from being state-of-the-art results. We expect to improve the results when combining the two types of features.

### 5.1.5 Combined Basic features

We have presented the results obtained when experimenting with acoustic and textual features separately. We now present the results obtained from performing the same experiments on combined text and audio data.

| **model** | **accuracy** | **f1-score** | **precision** | **recall** |
|---|---|---|---|---|
| Dummy | 0,400 | 0,143 | 0,100 | 0,250 |
| LR | 0,591 | 0,562 | 0,640 | 0,537 |
| MLP | 0,598 | 0,587 | 0,586 | **0,590** |
| SVC | 0,593 | 0,571 | 0,585 | 0,568 |
| RF | 0,482 | 0,415 | 0,477 | 0,411 |
| XGB | 0,574 | 0,546 | 0,585 | 0,530 |
| GNB | 0,547 | 0,557 | 0,589 | 0,562 |
| ENSEMBLE | **0,619** | **0,593** | **0,630** | 0,583 |

Table 5.5: Performance metrics for Combined basic features

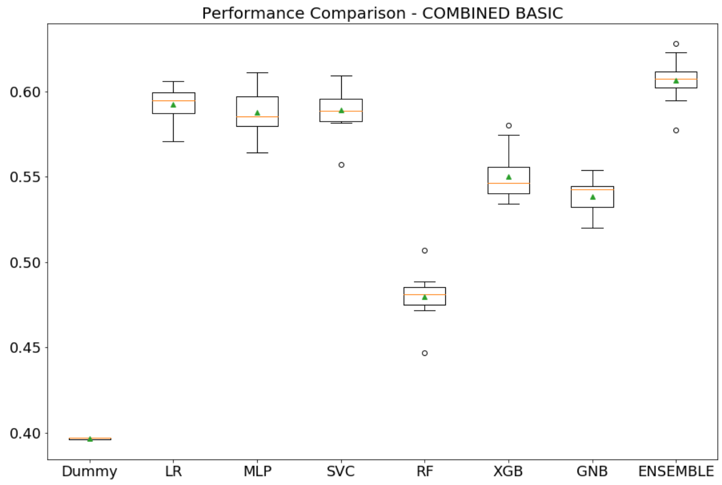


Figure 5.8: ML models performance over Combined Basic features

This experiment is conducted over textual TF-IDF features and acoustic Mel spectrogram features. In this case, the Ensemble model stacks the Logistic Regressor, Multilayer Perceptron and Support Vector Machine classifiers, resulting in the best performing model.

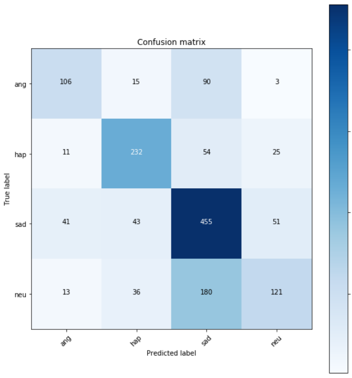


Figure 5.9: Ensemble Confusion matrix with combined basic features

Regarding the results in Section 5.1.4, this experiment demonstrates that textual and Mel spectrogram features perform slightly better when working together. The overall accuracy is increased over a 10% regarding the classification with only TF-IDF features (Table 5.4) but is not very representative comparing with only textual features.

Regarding the emotions classification, combining Mel spectrogram with text helps to improve the misclassification between sad and angry emotions in comparison with the textual features performance. However, the error between sad and neutral emotions remains intact.

### 5.1.6 Combined Hancrafted features

This experiment is conducted using TF-IDF and handcrafted features. Adding prosodic features has resulted in an improvement on performance regarding the experiment with only textual features, but it hasn't supposed an improvement in relation to the combination with Mel spectrogram features.

| model | accuracy | f1-score | precision | recall |
|---|---|---|---|---|
| Dummy | 0,379 | 0,137 | 0,095 | 0,250 |
| LR | 0,609 | 0,599 | 0,628 | 0,585 |
| MLP | 0,586 | 0,577 | 0,586 | 0,577 |
| SVC | **0,618** | **0,601** | **0,657** | 0,580 |
| RF | 0,535 | 0,475 | 0,613 | 0,473 |
| XGB | 0,566 | 0,533 | 0,593 | 0,518 |
| GNB | 0,542 | 0,556 | 0,585 | 0,571 |
| ENSEMBLE | 0,615 | 0,600 | 0,636 | **0,588** |

Table 5.6: Performance metrics for Combined Handcrafted features

In this case, the best performing model has been the Support Vector Classifier, achieving a 0.618 accuracy. The Ensemble model in this experiment is formed by the Linear Regressor, Multilayer Perceptron and Support Vector Classifier.

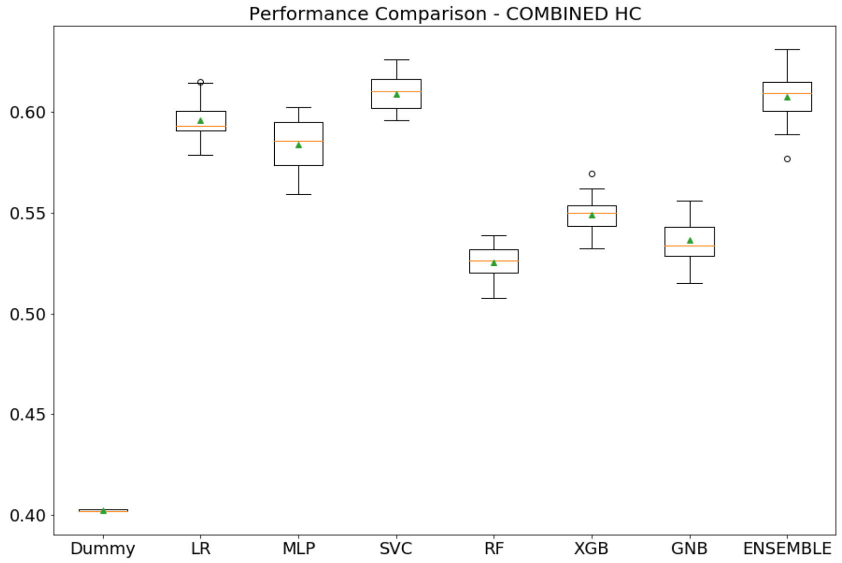


Figure 5.10: ML models performance over Combined Handcrafted features

In this case, the ensemble classifier clearly improves the performance of the most optimal models. Unlike in the case of textual features, when working with combined data, XGB, RF and MLP are between the best performing models.

### 5.1.7 Combined Extra features

As expected, the best results achieved with traditional ML classifiers have been produced by combining Mel, handcrafted and text features. In this case, we have improved the classification of emotions with only Mel features by a 14%. In this case, the Ensemble classifier combines the Support Vector, Logistic Regressor and Multilayer Perceptron classifiers.

| model | accuracy | f1-score | precision | recall |
|---|---|---|---|---|
| Dummy | 0,393 | 0,141 | 0,098 | 0,250 |
| LR | 0,610 | 0,568 | 0,654 | 0,548 |
| MLP | 0,624 | **0,613** | 0,623 | **0,606** |
| SVC | 0,615 | 0,589 | 0,641 | 0,572 |
| RF | 0,496 | 0,426 | 0,512 | 0,420 |
| XGB | 0,549 | 0,523 | 0,558 | 0,509 |
| GNB | 0,551 | 0,565 | 0,606 | 0,567 |
| ENSEMBLE | **0,631** | 0,597 | **0,670** | 0,582 |

Table 5.7: Performance metrics for Combined Extra features

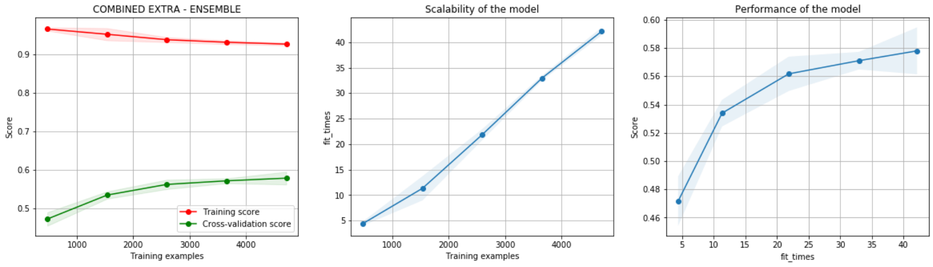


Figure 5.11: Learning curves of Ensemble model over combined extra features

Judging by the learning curves produced by the Ensemble classifier, it seems that the is a high bias and variance that could be improved by increasing the size of the training dataset.

## 5.2 Deep Neural Architectures

In section 5.1 we have reviewed the results when experimenting with different types of features, and concluded that the best performance was achieved when using Mel spectrogram and Handcrafted features together with textual TF-IDF features. We have also concluded that in some cases, the performance of traditional ML classifiers could be improved by increasing the size of our training dataset. However, the performance improvement wouldn't be big enough to be compared with state-of-the-art results.

In this section we present the results obtained by training the neural architectures described in Section 3.4.2, all trained with **Combined Extra features**.

### 5.2.1 DNN/BLSTM Results

The Dense-BLSTM Neural approach described in Section 3.4.2.1 improved the results achieved by ML classifiers. This approach has achieved a 0.6267 in testing accuracy. Table 5.8 contains the complete classification report. The network has been trained with 4 dense layers for audio input, with 512, 256, 128 and 32 hidden layers respectively. The text input has been fed to a BLSTM and a dense layer with 512 and 32 hidden layers respectively. Both flows have been merged using a concatenate layer followed by three additional dense layers with 1024, 512 and 4 hidden layers respectively.

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| **angry** | 0,647 | 0,538 | 0,588 | 221 |
| **happy** | 0,763 | 0,657 | 0,706 | 329 |
| **sad** | 0,576 | 0,829 | 0,680 | 580 |
| **neutral** | 0,626 | 0,315 | 0,419 | 346 |
| **accuracy** | | | | **0,627** |
| **macro avg** | **0,653** | **0,585** | **0,598** | **1476** |
| **weighted avg** | **0,640** | **0,627** | **0,611** | **1476** |

Table 5.8: Performance metrics for DNN/BLSTM

The training and validation curves (Fig 5.12 may suggest that we are using a number too large of features for training. While training validation achieves an 0.8 of accuracy, validation accuracy gets stuck around fit number 5 and validation loss hits its minimum around the 4th fit. After that, it starts overfitting and the validation loss starts increasing.

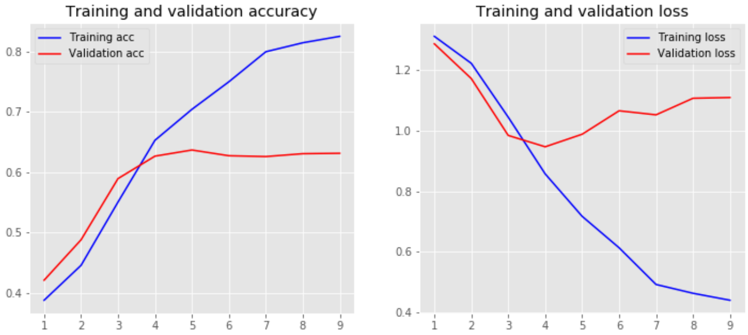


Figure 5.12: DNN/BLSTM Learning curves

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **angry** | 0.724 | 0.574 | 0.640 | 221 |
| **happy** | 0.778 | 0.760 | 0.769 | 329 |
| **sad** | 0.664 | 0.815 | 0.731 | 580 |
| **neutral** | 0.583 | 0.450 | 0.508 | 346 |
| **accuracy** | | | | **0,681** |
| **macro avg** | **0.687** | **0.650** | **0.662** | **1476** |
| **weighted avg** | **0.679** | **0.681** | **0.673** | **1476** |

Table 5.9: Performance metrics for CBL/BLSTM

### 5.2.2 CBL/BLSTM Results

This experiment is based on the neural architecture described in Section 3.4.2.2. This network feeds the audio input into a Bi-directional LSTM and a Convolutional neural network, and the text input into a single BLSTM. The CNN for audio is formed by two convolutional layers collated with one MaxPooling layer each (with 64 and 32 hidden layers respectively). The audio BLSTM output is merged with the CNN audio output and the BLSTM text output.

In this experiment we observe an improvement on performance compared to the previous one. In order to avoid overfitting, the number of TF-iDF features is reduced. When building the vocabulary, we have ignored terms that have a document frequency strictly higher than a threshold set to 10. Still, the validation learning curve comes to a standstill around epoch number 6 and stops increasing. This may indicate that further feature selection could be needed.

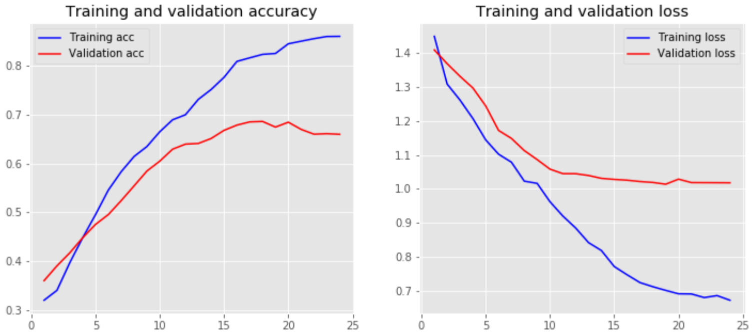


Figure 5.13: CBL/BLSTM Learning curves

### 5.2.3 CBLA/BLSTM Results

In this experiment we have tested the same architecture than in Experiment 5.2.2, but with the addition of an attention layer with 64 hidden layers. This layer is added after the BLSTM layer for audio, and later merged with the CNN audio and BLSTM text flows.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **angry** | 0,647 | 0,548 | 0,593 | 221 |
| **happy** | 0,701 | 0,669 | 0,684 | 329 |
| **sad** | 0,558 | 0,800 | 0,657 | 580 |
| **neutral** | 0,608 | 0,251 | 0,356 | 346 |
| **accuracy** | | | | **0,604** |
| **macro avg** | **0,628** | **0,567** | **0,573** | **1476** |
| **weighted avg** | **0,615** | **0,604** | **0,583** | **1476** |

Table 5.10: Performance metrics for CBLA/BLSTM

The learning curves for this experiment show a low bias case, where adding additional data wouldn't help. The component of attention doesn't seem to add any improvements in terms of accuracy, but the model seem to learn more gradually. This may suggest that the attention component may have a good potential but the architecture of the network is not complex enough.

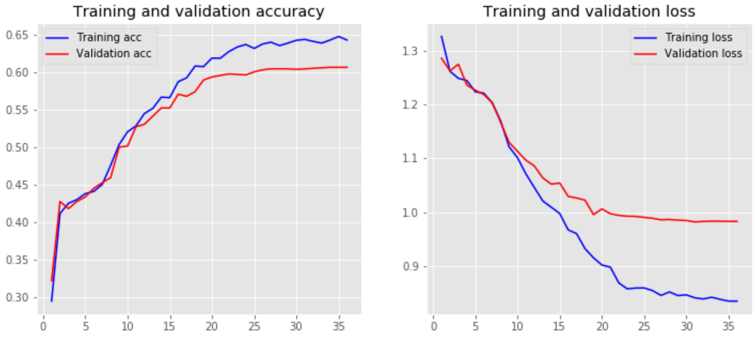


Figure 5.14: CBLA/BLSTM Learning curves

### 5.2.4 Summary of results

The following table presents the results achieved by the best ML models and each Neural Network for all the experiments explained above.

| **method** | **model** | **accuracy** | **f1-score** | **precision** | **recall** |
|---|---|---|---|---|---|
| Audio Basic | XGB | 0.493 | 0.441 | 0.490 | 0.432 |
| Audio HC | RF | 0.455 | 0.401 | 0.448 | 0.404 |
| Audio Extra | ENSEMBLE | 0.497 | 0.451 | 0.506 | 0.437 |
| Text only | SVC | 0.602 | 0.566 | 0.627 | 0.547 |
| Combined Basic | ENSEMBLE | 0.619 | 0.593 | 0.630 | 0.583 |
| Combined HC | SVC | 0.618 | 0.601 | 0.657 | 0.580 |
| Combined Extra | ENSEMBLE | 0.631 | 0.597 | 0.670 | 0.582 |
| Combined Extra | DNN/BLSTM | 0,627 | 0,611 | 0,640 | 0,627 |
| Combined Extra | **CBL/BLSTM** | **0,681** | **0,662** | **0,687** | **0,650** |
| Combined Extra | CBLA/BLSTM | 0,604 | 0,583 | 0,615 | 0,604 |

Table 5.11: Summary of results by all classifiers

# Chapter 6

# Conclusions and future work

In this chapter we summarize the work developed during this Thesis and detail the most notable conclusions that we have drawn. Later on we present some lines for future research.

## 6.1 Conclusions

Despite the lately progresses made in the field of Speech Emotion Recognition, it remains a challenging task. The multimodal approaches to solve this task are still very immature in terms of methodology. Even though there are sophisticated methods that aim to tackle this task ([Li et al., 2019], [Yoon et al., 2018]), there is no established criteria for choosing the best features to study. Emotion recognition is, by definition, an extremely ambiguous task even for humans. Unlike other tasks related to Natural Language, there isn't a specific architecture that seems to work specially better when it comes to emotion recognition.
Another limitation regarding SER is certainly the lack of quality, available data to build systems upon. Despite that IEMOCAP provides a good quality of annotated utterances, the volume of data remains too low to achieve great results.

Regarding the observed constraints in SER, in this work we have **defined an end-to-end pipeline for multimodal recognition**, using a wide variety of features and we have assessed their contributions and interactions between each other. With the goal of comparing the potential of classical Machine Learning and Deep Learning methods for this specific task, we have made **an study using different types of features and evaluated their performances**. Finally, we have developed an application for retrieving new audio and text data, allowing us to **keep retraining and improving a model**. This way, we have tackled the problem of the amount of available data. This application can be used in a future to keep retrieving utterances labeled with emotions.

Having extracted three types of different features (mel-spectrogram, prosodic and TF-IDF features) from both audio and text, we have observed that **our experiments improved when using the combination of all these features together**. In the case of some classifiers, they appeared to be overfitted when using mel-spectrogram features. We concluded that this could be due to the great number of features that mel spectrogram contains and could be avoided with further feature selection. On the other hand, **mel features and prosodic features didn't contribute much with each other** when performing classification with **only audio features**, but **jointly improved the emotion classification when using text**.

The deep neural approaches have resulted to perform slightly better than the classical ML classifiers, but with no major difference. We have observed that there is no clear correlation between the complexity of the network used and its results. DNN and CBL worked better than when using an attention mechanism. The main conclusion that we extract from this experiment is that DL methods seem to have better potential and capacity for improvement than classical ML models for multimodal tasks, as long as their architectures remain simple enough when using audio and text inputs.

The error analysis made in all our experiments shows that the highest confusion rate appears always between "Neutral" and "Sad" emotions. We conclude that this can be due to a problem of class imbalance (having much more "Sad" than "Neutral" samples) but also emotion ambiguity. We consider class imbalance as a particular problem of lack of training data which can be solved increasing the training set. Based on this premise, the bot developed to retrieve feedback from classifications appears to be a good solution that allows a future continuous retraining and data retrieval.

## 6.2 Future work

Working with text and audio data simultaneously is not a trivial task. In this work we have combined these two types of information without giving an special weight to any feature. In other words, a spectral component of and audio sample had the same importance as one word. A future line for research could be focused on defining the best way to combine this information into a weighted set of features with relative importances.

Also, since two of the major limitations in this work have been the lack of a larger training set and a class imbalance problem, we propose data augmentation as a future research direction. One possible approach for this line of research could be the generation of synthetic audio samples using an algorithm such as SMOTE [Chawla et al., 2011] that would contribute on improving the overall performance of the system.

Another future work could be focused on studying the feature selection mechanisms suitable for high-dimensional features such as the mel spectrogram. The selection of spectral features could help us on focusing on the frequency components that bring most information for each audio sample. Regarding textual information, different methods than TF-IDF such as word embeddings, could be included in this study.

# Bibliography

Al-Talabani, A., Sellahewa, H., and Jassim, S. A. (2015). Emotion recognition from speech: tools and challenges. In *Mobile Multimedia/Image Processing, Security, and Applications 2015*, volume 9497, pages 193 – 200. International Society for Optics and Photonics, SPIE.

Alpert, M., Pouget, E., and Silva, R. (2001). Reflections of depression in acoustic measures of the patient's speech. *Journal of affective disorders*, 66:59–69.

Badshah, A., Ahmad, J., Rahim, N., and Baik, S. (2017). Speech emotion recognition from spectrograms with deep convolutional neural network. pages 1–5.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Bandela, S. R. and Kumar, T. K. (2017). *Stressed speech emotion recognition using feature fusion of teager energy operator and MFCC.*

Besson, M., Magne, C., and Schön, D. (2002). Emotional prosody: sex differences in sensitivity to speech melody. *Trends in Cognitive Sciences*, 6(10):405–407.

Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49 – 59.

Buechel, S. and Hahn, U. (2017). Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2011). Smote: Synthetic minority over-sampling technique.

Devillers, L., Vidrascu, L., and Lamel, L. (2005). Challenges in real-life emotion annotation and machine learning based detection. *Neural networks : the official journal of the International Neural Network Society*, 18(4):407—422.

Douglas-Cowie, E., Campbell, N., Roach, P., and Cowie, R. (2003). *Speech Communication*, 40(1-3)(1-2):33–60.

Ekman, P., Sorenson, E. R., and Friesen, W. V. (1969). Pan-cultural elements in facial displays of emotion. *Science*, 164(3875):86–88.

Emerson, C., Harrison, D., and Everhart, D. (1999). Investigation of receptive affective prosodic ability in school-aged boys with and without depression. *Neuropsychiatry, neuropsychology, and behavioral neurology*, 12:102–9.

Gideon, J., Khorram, S., Aldeneh, Z., Dimitriadis, D., and Provost, E. M. (2017). Progressive neural networks for transfer learning in emotion recognition. In *Proc. Interspeech 2017*, pages 1098–1102.

Han, K., Yu, D., and Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In *Interspeech 2014*.

Kim, J., Englebienne, G., Truong, K. P., and Evers, V. (2017). Towards speech emotion recognition "in the wild" using aggregated corpora and deep multitask learning.

Li, Y., Zhao, T., and Kawahara, T. (2019). Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning. In *INTERSPEECH*.

Sahu, G. (2019). Multimodal speech emotion recognition and ambiguity resolution. *ArXiv*, abs/1904.06022.

Schuller, B., Rigoll, G., and Lang, M. (2004). Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. volume 1, pages I – 577.

Seehapoch, T. and Wongthanavasu, S. (2013). Speech emotion recognition using support vector machines. *2013 5th International Conference on Knowledge and Smart Technology (KST)*, pages 86–91.

Tisljár-Szabó, E. and Pléh, C. (2014). Ascribing emotions depending on pause length in native and foreign language speech. *Speech Communication*, 56:35–48.

Ververidis, D. and Kotropoulos, C. (2012). A state of the art review on emotional speech databases.

Volkmann, J., Stevens, S. S., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):208–208.

Yoon, S., Byun, S., and Jung, K. (2018). Multimodal speech emotion recognition using audio and text.

Zhang, B., Khorram, S., and Mower Provost, E. (2019). Exploiting acoustic and lexical properties of phonemes to recognize valence from speech. pages 5871–5875.