ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

LARISSA DAIANE CANEPPELE GUDER

DIMENSIONAL SPEECH EMOTION RECOGNITION FROM BIMODAL FEATURES

Porto Alegre
2024

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

DIMENSIONAL SPEECH EMOTION RECOGNITION FROM BIMODAL FEATURES

LARISSA DAIANE CANEPPELE GUDER

Master Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Dalvan Griebler
Co-Advisor: Prof. João Paulo Aires

Porto Alegre
2024

Dedicated to Cristina and Maria

“9iooooooo89ttttttttt/kl,7-=]/mfffffrt/7f/, miiiii-iiiiiiiiiiiiiiiiiiii fffffffffffffffdrr rrrrrrrrrrrrr rrrrrrrrrr roooooooooooooo-=”
(Pumba)

ACKNOWLEDGMENTS

À lorem ipsum, dolor sit amet consetetur sadipscing elitr sed diam...

RECONHECIMENTO DIMENSIONAL DE REMOÇÕES NA FALA ATRAVÉS DE INFORMAÇÕES BIMODAIS

RESUMO

Seu resumo em português aqui. lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita. kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at.

FRM: O abstract deve conter quatro pontos principais: 1. Qual o propblema; 2. Por que este problem importa e/ou é difícil; 3. Como tu resolves ele; e 4. Que consequência tem isto?

Palavras-Chave: lorem, ipsum, dolor, sit, amet.

DIMENSIONAL SPEECH EMOTION RECOGNITION FROM BIMODAL FEATURES

ABSTRACT

Your abstract in English here. lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at

Keywords: lorem, ipsum, dolor, sit, amet.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

LIST OF ABBREVIATIONS

LIST OF SYMBOLS

CONTENTS

1. INTRODUCTION

FRM: Overall this text is pretty decent, so kudos for your current work. Having said that, I have made a series of comments below that I hope will make this text even better. If you have time to check this out, read the points I wrote here: http://www.meneguzzi.eu/felipe/writing.shtml

FRM: What the hell is the paragraph below?

Copy/paste, mas o termo idiossincrática é maravilhoso (é um adjetivo que se refere à idiossincrasia, que é a maneira de ver, de sentir e de reagir, própria de cada pessoa.) These works would allow to overcome the shortage of data and learm how to build more general recognizers, that are able to recognice emotions across the diversity of **idiosyncratic** features of the various databases. de Lope and Graña (2023)

- Why emotions, importance, context, historical maybe
- Where it can be applied in the real world
- How emotions are measured by computers
- Bimodal approaches
- Streaming factor
- What I did
- Section structure

Emotion recognition is much more a perspective than an exact science. Besides the ways used to classify emotions in psychology, ~~two approaches were adopted by the computer science area~~computer science adopts two approaches to classify emotions: discrete classes and dimensional ones. In discrete classes, we have six emotions considered essential by Ekman (1999), defined as anger, disgust, fear, happiness, sadness, and neutral. On the other hand, Russell (1980) defines a dimensional approach through the circumplex model of affect. With the circumplex, we can reach a specific emotion through two dimensions: arousal and valence. Arousal is related to calming or exciting the tonality of speech, while valence represents how pleasant or not it is. ~~In complement,~~ Mehrabian (1996) adds the dominance dimension, which represents how emotion influences a person's behavior. The dimensional approach allows for a flexible number of emotions in the model.

Emotion recognition has been applied in a large set of research application areas. ~~The review by~~ Geetha et al. (2024) identifies sectors like education, healthcare, marketing and advertising, human-robot interaction, security and surveillance, customer service, sports, entertainment, gaming, and the automotive industry. Conversely, the preoccupation with privacy and the possible emotional state exploration to induce the user to buy some services or products is discussed by Testa et al. (2023).

FRM: Two things here: 1. Avoid passive voice; 2. You need a citation tu underpin both statements (that CS classifies emotions differently than psychology, and that these two are what CS uses).

FRM: Is this the definition of the circumplex model?

FRM: In complement does not work well in English

FRM: No need to add that fluff

The lack of large amounts of data for training and testing deep learning models makes it difficult for the field of SER to grow. Existing datasets either have a small amount of available data, are less diverse than necessary, or are too different from real-world data. Even with such a small amount of data, we have new approaches that use deep learning models with text and acoustic features. Some authors have already proven that using text features, such as word embeddings, improves valence prediction (Triantafyllopoulos et al., 2022; Srinivasan et al., 2022; Ghriss et al., 2022; Atmaja and Akagi, 2020, 2021; Sogancioglu et al., 2020; Julião et al., 2020). However, including new features in the processing usually increases the time necessary to generate an output. For instance, the inclusion of text features makes it necessary to first transcribe the audio to use it as input.

FRM
Prove is a very strong word in CS. Only use "prove" when you have a mathematical proof that something is the case. Perhaps "shown"?

~~Considering these challenges and open gaps in the literature, this master thesis presents an architecture for speech emotion recognition that can be used in a streaming environment.~~This work aims to address these challenges through an architecture for speech emotion recognition useable in a streaming environment.

FRM
How about this?

FRM: The content below, and the next paragraph would read better if you cast in terms of your contributions. For example:
"Our key contributions are twofold. First, we investigate the current state of the art in emotion recognition and identify key gaps in current methods. Second we develop as series of software architectures that overcome these gaps via X, Y, Z (here you can reuse your content). We empirically show the effectiveness..."

To define this architecture, we first explore the use of hand-crafted audio features and audio embeddings for speech emotion recognition; sentence embedding models for emotion recognition in the text; and pre-trained models for automatic speech recognition. The evaluation of these aspects considers the time necessary to extract and process the features, the Mean Squared Error (MSE) metric for emotion recognition, and the Word Error Rate (WER) for automatic speech recognition.

Our final architecture uses the WhisperX model to perform the automatic speech recognition task. The representation and audio representation are made with Mini LM L3 and VGGIsh respectively. To predict the arousal, valence, and dominance values, we use an LSTM network. Before sending the input to the LSTM, we apply the PCA algorithm to Mini LM L3 to reduce the dimensionality to the same size as VGGIsh embedding. After that, we use a concatenation layer that will join the features. In contrast to Atmaja and Akagi (2020) that uses word embeddings and hand-crafted audio features and achieves 0.571 of CCC for arousal, 0.418 for valence, and 0.500 for dominance, we achieve 0.5915 of CCC for arousal, 0.4165 for valence and 0.5899 for dominance.

This master thesis is divided into four chapters: first, we have the Background on Chapter 2, where we discuss the main concepts about affective computing, audio processing, sentence representation, recurrent neural network, speech emotion recognition, and data streaming. Then, in Chapter 3, we discuss some related works and the main differences from our proposal. Our main findings are presented in Chapter 4, where we discuss our final architecture and compare the results with state-of-the-art approaches. Finally, in Chapter 5, we present the necessary experiments to define the representations and architecture.

2. BACKGROUND

2.1 Emotions

Besides the human perception of emotions, to make possible to a computer recognize emotions, we have some research areas focusing on this. Created by Picard (1997), the terminology “affective computing” defines the research focus on recognizing, interpreting, and influencing human emotions through the use of technology. From Affective Computing, new approaches emerged focusing on different understandings of the human behaviour. We have different levels of approaches that focus on understanding human behavior through the recognition of emotions and analysis of sentiments (Wang et al., 2022).

When dealing with the definition terms presents in affective computing, it is necessary to notice their different meanings. There is a substantial difference between emotions and sentiments, as shown in Figure 2.1. Munezero et al. (2014) defines the difference between affect, feelings, emotions, sentiments, and opinions. First, affect is a non-conscious phenomenon, while feelings are person-centered consciousness and represent the expression of affect. Emotions are influenced by social and cultural factors and represent the preconscious expression of feelings and affect. Otherwise, sentiments are conscious and built over time, considering social influence. Finally, the opinion is more related to how each person interprets information, considering or not emotions.

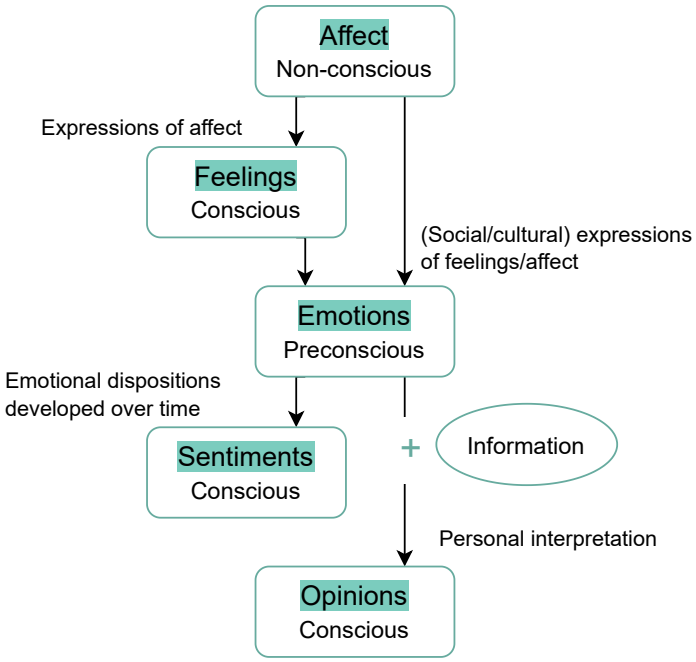

Figure 2.1 – Differentiating factors between affect, feelings, emotions, sentiments, and opinions. Adapted from Munezero et al. (2014)

Emotions are complicated; expressing or understanding what we feel is sometimes difficult. The definition of an emotion is not a static thing. Boehner et al. (2005) defines that emotions are culturally grounded and dynamically experienced, to some extent, constructed in interaction. Complementary to that, Loderer et al. (2020) states that emotions can be perceived differently in different cultures, but even cross-cultural approaches are guaranteed.

Understanding what the other is feeling is one of the bases of relationships, whether in society or the family environment. From an evolutionary perspective, emotions directly impact our sense of survival. For example, fear is an essential regulator that can help in decision-making. Generally, the demonstration of emotion occurs naturally and subconsciously. They can be perceived by facial and corporal expressions, vocal intonation, pupil dilatation, heart rate, and breathing (Picard, 1997).

Figure 2.1 details the structure of an emotion. At the top, we have the culture-independent level, which represents how the individual perceives emotions, Then, we have the culture and society-dependent level, which describes the way individuals name how they feel. From this perspective, to make it possible for a computer to recognize an emotion, it is necessary to classify it mathematically.

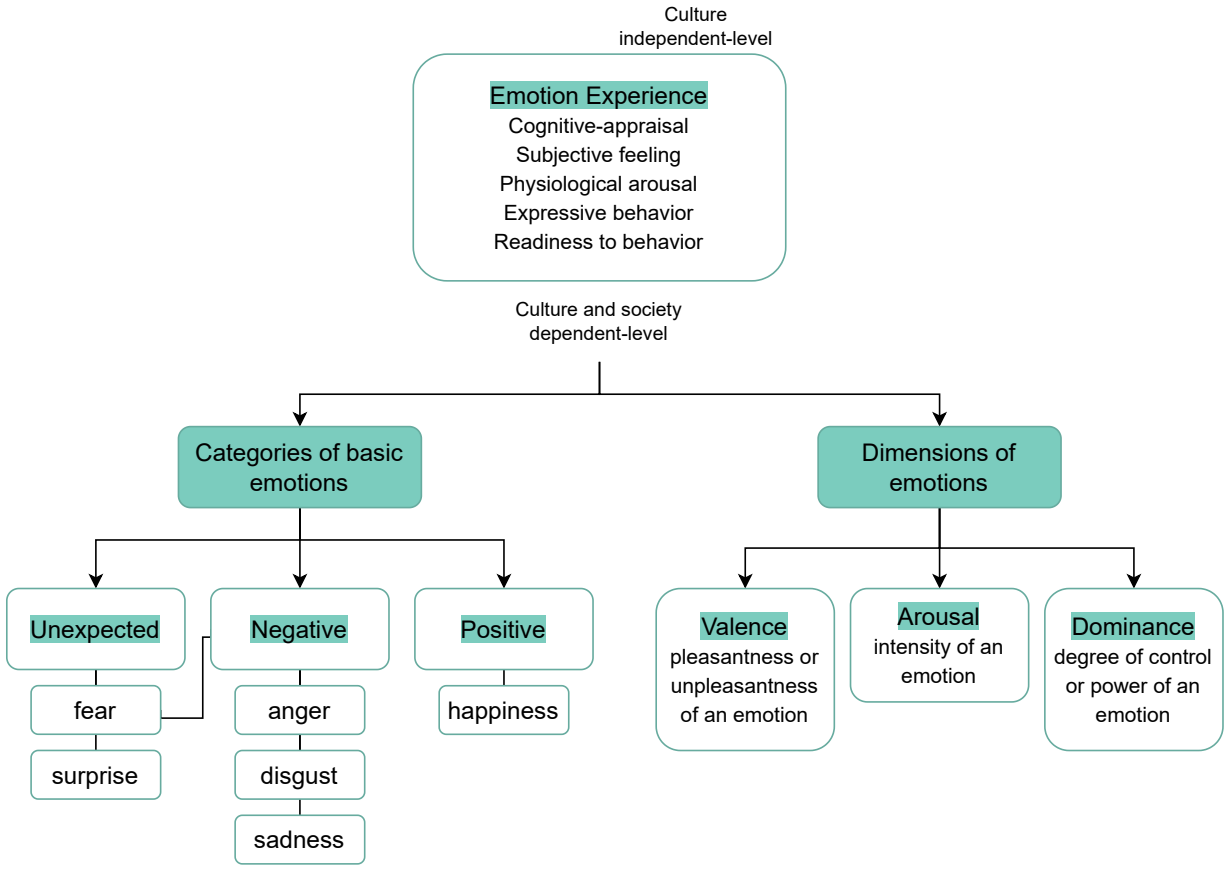

Figure 2.2 – Structure of emotion experience and classification, adapted from Munezero et al. (2014) and Roberts et al. (2012)

As the psychology literature explains, two different classifications are often used for this purpose: discrete classification and dimensional classification. Ekman (1999) proposes what we will call the discrete classification of emotions. His study is an update from a previous work published in 1957. He proposes six basic emotions: anger, disgust, fear, happiness, sadness, and surprise. Anger, disgust, fear, and sadness represent all negative emotions, happiness represents positive emotions, and fear and surprise represent unexpected emotions.

On the other hand, we have the circumplex model of affect proposed by Russell (1980), which we will call dimensional classification. Two axes represent values for arousal (y-axis) and valence (or pleasure) (x-axis) in a dimensional space. With these two values that range from -1 to 1, it is possible to determine emotion. The arousal is related to the acoustic features, and valence is related to the linguistic features. Figure 2.3 presents an updated version proposed by Scherer (2005), which has more emotions mapped than the original version proposed by Russell (1980). Each plus(+) sign refers to an emotion's exact point in space.

In the circumplex model, we can see that we obtain happy and excited emotions with high valence and arousal values. Feelings like gladness and calm can be found when the valence is low and arousal high. Sad, tired, and bored emotions are related to low valence and arousal values, while we have frustration, anger, and fear emotions for high valence and low arousal.

Complementing the Circumplex model, Mehrabian (1996) proposed the Pleasure-Arousal-Dominance Emotional State Model, represented in Figure 2.1. In addition to valence (pleasure) and arousal dimensions, we have dominance as a third dimension. Dominance refers to how emotion influences a person's behavior. Lower levels represent passive or submissive feelings, while high levels are assertive or powerful.

Besides the different ways to express emotions, two main areas of research focus on vocal intonation: the first one is trying to recognize emotions from speech, and the second is making it possible for a computer to synthesize audio with emotions. To understand how a computer can recognize emotions from speech, first, it is necessary to understand how the audio signal is processed. We discuss it in Section 2.2.

2.2 Audio Processing

One of the bases of human communication is speech. Through speech, we can transmit information and express our emotions. Audio processing is a subfield in Digital Signal Processing that aims to convert sound into a format that allows machines to process it. Huang et al. (2001) define sound as a "longitudinal pressure wave formed of compressions and rarefactions of air molecules, in a direction parallel to that of the application of energy."

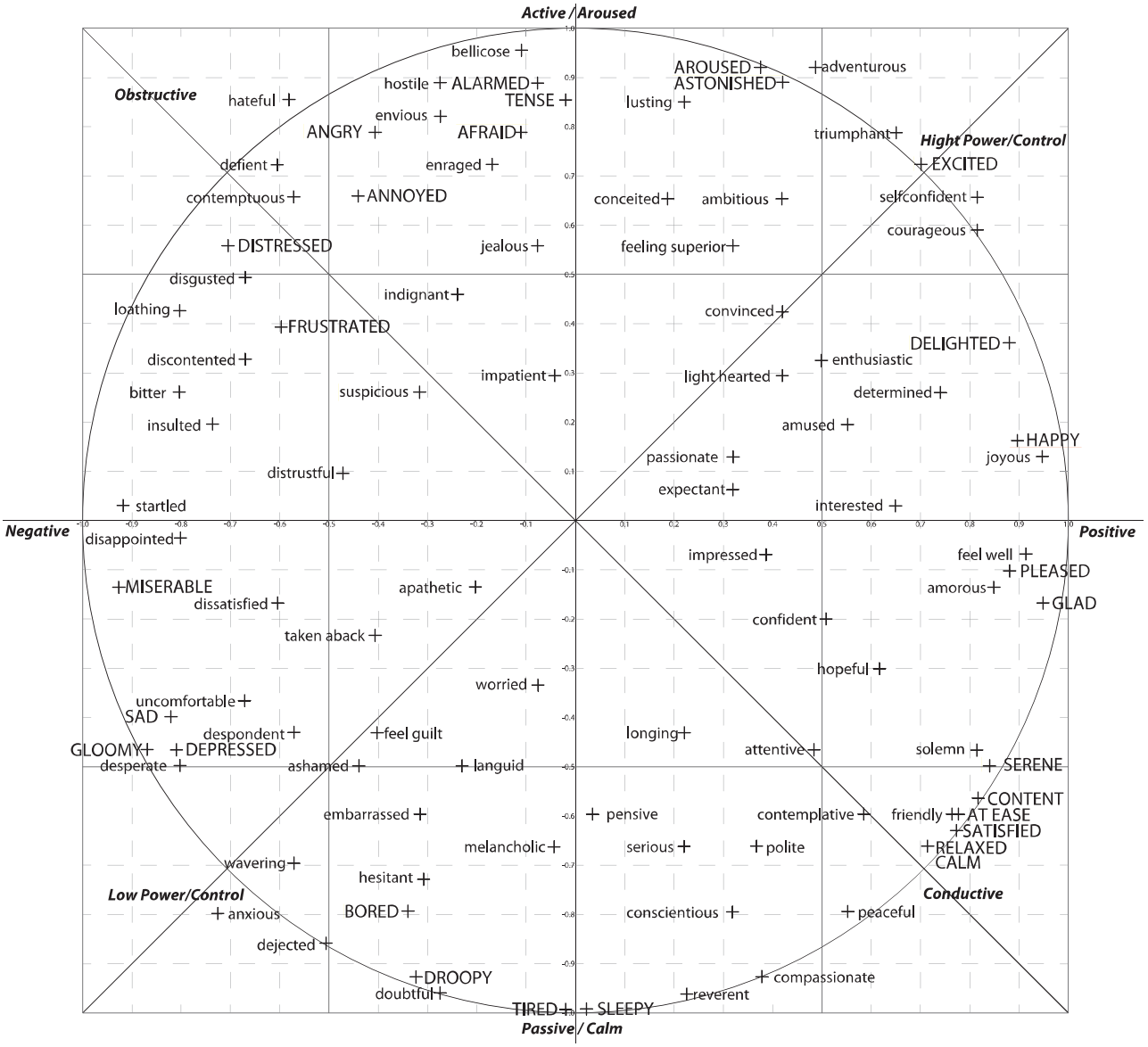

Figure 2.3 – Updated version from Russell (1980) circumplex model of affect, proposed by Scherer (2005), focusing on the semantic space for emotions. Adapted from Ahn et al. (2010).

In computing, audio processing involves, in the first place, converting the analog signal to digital. Two process stages are necessary to make signal conversion possible: sampling and quantization. While sampling reduces continuous-time signals to discrete-time signals, quantization is responsible for converting the signal from continuous to discrete (Smith, 1997).

To make it possible to recognize emotions in speech, first, we need to extract features from the low-level descriptors (LLDs). LLDs provide ways to extract information from the digital signal. They can be grouped into three domains: prosodic, spectral, and voice quality. Besides the full set of features, for emotion recognition in the prosodic domain, there are three most commonly used: fundamental frequency or pitch, energy, and duration. The Fundamental Frequency, also called F0, opens and closes the vocal folds

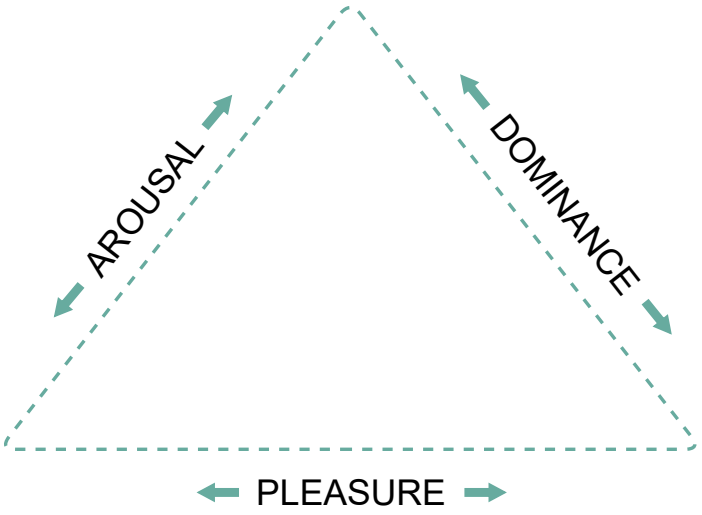

Figure 2.4 – Pleasure-Arousal-Dominance Emotional State Model proposed by Mehrabian (1996)

in phonation (Huang et al., 2001). Williams and Stevens (1972) described the positive impact of using the F0 feature for SER tasks in 1972, and modern approaches, such as the ones proposed by Atmaja and Akagi (2020), MacAry et al. (2021), and Julião et al. (2020) still use it. Energy represents how loud or intense a sound signal can be, it is measured in decibels (dB). Intensity is directly related to the arousal dimension. Duration represents the time that a sound or syllable is produced.

Regarding spectral features, the most commonly used feature is Mel-Frequency Cepstral Coefficients (MFCCs). Regarding spectral features, the most commonly used feature is Mel-Frequency Cepstral Coefficients (MFCCs). Huang et al. (2001) define MFCCs as a representation of a cepstrum of a determined windowed short-time signal. This representation is derived from the Fast-Fourier-Transformation (FFT) of that signal. Log Mel Spectrogram can represent audio signals in the frequency domain. MFCCs are used to translate for a machine how humans perceive sounds.

Figure 2.5 illustrates the steps MFCC takes to generate the representation. First, we have the application of a window function, like a hamming window, in each frame of the signal. Afterward, an FFT is applied to transform the signal to a frequency domain. As the frequency is measured in HZ, it is necessary to convert it into a Mel Scale, so a filter bank is used. A filter bank is necessary because the human voice spectrum is not linearly distributed (Brunet et al., 2013). They also contribute to capturing the essential characteristics of the task that will be performed. After that, a log compression of the Mel scale using a natural logarithm transforms the signal into something more related to how humans perceive sounds. Finally, the Discrete Cosine Transform (DCT) converts the log-compressed Mel-scaled into the cepstral domain, the MFCC.

And for voice quality features, we have shimmer and jitter. Related to voice quality features, we have shimmer and jitter. Jitter is calculated through Equation 2.1, where

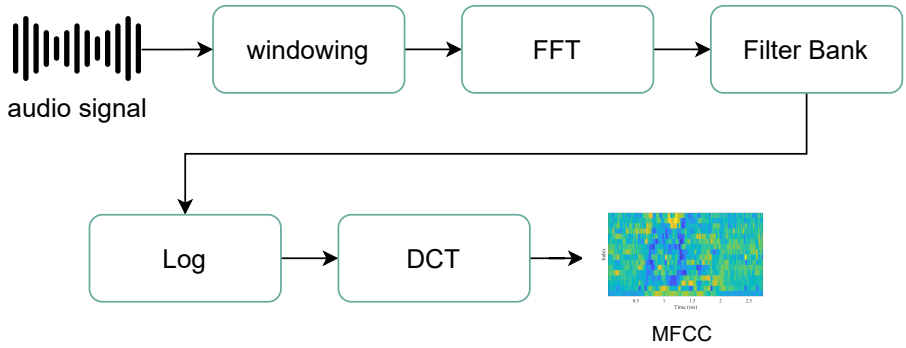

Figure 2.5 – MFCC process

T_i is the considered pitch period, and N is the number of cycles. Jitter represents the variation of the F0 over time. These variations depend on many factors and are directly related to the emotional state of the speaker (Koolagudi et al., 2018).

$$Jitter = \frac{1}{N-1}\sum_{i=1}^{N-1}|T_i - T_{i+1}| \tag{2.1}$$

On the other hand, shimmer can calculate the energy variation. Koolagudi et al. (2018) defines shimmer as "the representation of variation in the amplitude between adjacent F0 periods". Shimmer Equation 2.2 is composed of the extracted peak-to-peak amplitude data A_i and the number of F0 periods N.

$$Shimmer = \frac{1}{N-1}\sum_{i=1}^{N-1}|20\log\left(\frac{A_{i+1}}{A_i}\right)| \tag{2.2}$$

In the next sections, we explain existing hand-crafted features that can be extracted using specific libraries; and audio embeddings, produced by machine learning models based on audio features.

2.2.1 Handcrafted features

The process of extracting features using existing libraries is called handcraft feature extraction. In the literature, some standards of specific feature sets are being used for SER. The most common ones are eGeMAPs (Eyben et al., 2016) and ComParE (Schuller et al., 2016). eGeMAPs is the extended version of GeMAPs (Geneva Minimalistic Acoustic Parameter Set), which contains 88 parameters, such as frequency, energy/amplitude, and spectral (balance/shape/dynamics) features. On the other hand, ComParE provides 6,373 features composed by the statistical functionals over the LDDs.

To extract these feature sets from audio, there are two main Python libraries: OpenSmile (Eyben et al., 2010) and pyAudioAnalisys (pAA) (Giannakopoulos, 2015). Using

OpenSmile, we can extract eGeMAPs and ComParE feature sets. We detail the complete feature groups and descriptions that can be extracted using OpenSmile in Table 2.2.1. In addition, at feature-level, it is possible to extract the feature sets using three different approaches: (1) only the LLDs, which are calculated over a sliding window; (2) Delta regression of LLDs and (3) the statistical functionals, which maps LLDs values to static values (Eyben et al., 2010).

Feature Group	**Description**
Waveform	Zero-Crossings, Extremes, DC
Signal energy	Root Mean-Square & logarithmic
Loudness	Intensity & approx. loudness
FFT spectrum	Phase, magnitude (lin, dB, dBA)
ACF, Cepstrum	Autocorrelation and Cepstrum
Mel/Bark spectr	Bands 0-Nmel
Semitone spectr.	FFT based and filter based
Cepstral	Cepstral features, e.g. MFCC, PLPCC
Pitch	F0 via ACF and SHS methods Probability of Voicing
Voice Quality	HNR, Jitter, Shimmer
LPC	LPC coeff., reflect. coeff., residual Line spectral pairs (LSP)
Auditory	Auditory spectra and PLP coeff.
Formants	Centre frequencies and bandwidths
Spectral	Energy in N user-defined bands, multiple roll-off points, centroid, entropy, flux, and rel. pos. of max./min
Tonal	CHROMA, CENS, CHROMAbased features

Table 2.1 – openSMILE's low-Level descriptors, extracted from Eyben et al. (2010)

pAA is an open-source option for extracting features. Table 2.2.1 shows the complete features and descriptions from short-term extraction. From the features detailed in the table, it is possible to extract them using two functions: one for short-term features and another for mid-term features. The short-term features use windowing to split the signal into frames and process the features for each frame. This method extracts a total of 34 features. With the mid-term features, it is possible to extract the mean and standard deviation for each short-term feature. Using the mid-term feature extraction, the total number of features is 136.

2.2.2 Audio Embedding

Besides using handcrafted features, it is possible to use audio embeddings to recognize emotions. Two of the existing alternatives for that are: TRILL (Shor et al., 2020) and VGGish (Hershey et al., 2017). The TRIpLet Loss network (TRILL) is a self-supervised model trained on the AudioSet dataset, created focusing on non-semantic tasks (that do not consider the meaning or the presence of the words in the speech). The architecture

ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Table 2.2 – PAA Features, extracted from Giannakopoulos (2015)

uses a variant of the ResNet-50 with a 512-dimensional embedding layer (Shor et al., 2020).

VGGish (Hershey et al., 2017) is a modification of the VGG16 architecture (Simonyan and Zisserman, 2015), a popular convolutional neural network. The authors trained the VGGish model on a large YouTube dataset. The input of the VGGish is a numerical representation of the audio waveform. This audio can have 10 seconds as the maximum length of duration The output of the VGGish is an audio embedding representation with 128 dimensions.

2.2.3 Automatic Speech Recognition

The Automatic Speech Recognition (ASR) task aims to convert audio signals captured from speech into text, according to the language of the speaker (Goodfellow et al.,

2016). Malik et al. (2021) define a standard model architecture consisting of four steps, as we can see in detail in Figure 2.6. In this architecture, after getting the input sound wave, the first step is a preprocessing module to clear the audio input, remove unwanted noises, and prepare it for the feature extraction step. The most common techniques used in pre-processing are voice activity detection, noise removal, pre-emphasis, framing, windowing, and normalization (Labied et al., 2022).

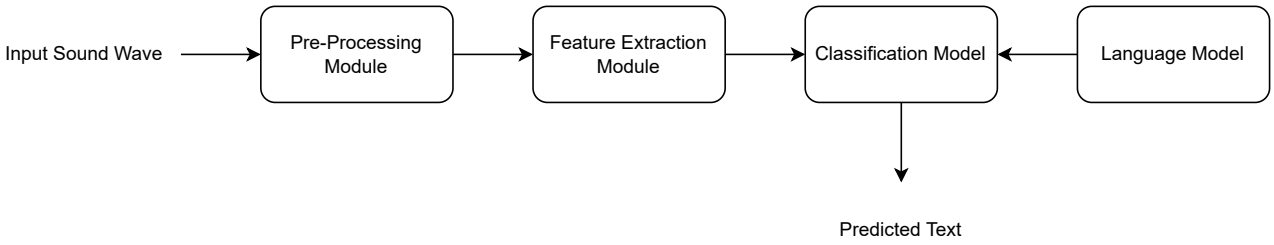

Figure 2.6 – Basic structure of an ASR, adapted from Malik et al. (2021)

After cleaning the audio input, it is necessary to extract the input features for the model. Basically, there are two feature domains: spectral and temporal. Temporal features are based on the time domain, while spectral features are based on the frequency domain. For ASR, it is common to use MFCC, PLP, DWT, relative spectral-perceptual linear prediction (RASTA-PLP), and LPC (Malik et al., 2021).

When feature extraction is done, a classifier uses the features to predict what was spoken on the audio. The most commonly artificial neural networks used are Multi-Layer Perceptron (MLP), Self-organizing maps (SOM), Radial Basis Functions (RBF), Recurrent neural network (RNN), Convolutional neural network (CNN), Fuzzy neural network (FNN), and Support vector machines (SVM) (Malik et al., 2021).

To evaluate the classifier, the typical metrics used are Word Error Rate (WER) and Character error rate (CER). These metrics focus on identifying the percent of wrong predictions regarding words and characters, where the perfect result is 0. The equation structure of these metrics is the same, as defined in Equation 2.3. We have the sum of the number of substitutions S, deletions D, and insertions I divided by the number of elements N in the ground truth. Substitutions are related to the number of characters/words that are either different or in a different position from the original sentence. Deletions are the number of characters/words removed from the original sentence to reach the original sentence. And, finally, insertions are related to the extra characters/words necessary to obtain the correct sentence.

$$WER = \frac{S + D + I}{N} \qquad CER = \frac{S + D + I}{N} \tag{2.3}$$

Surveys such as the one conducted by Roger et al. (2022) list the most used datasets in the ASR task. The datasets are: LibriSpeech (Panayotov et al., 2015), IEMO-CAP (Busso et al., 2008), and VoxCeleb1 (Nagrani et al., 2017). The state-of-the-art models

in these datasets are: Pase+ (Ravanelli et al., 2020), Wav2Vec2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021) and AutoSpeech (Ding et al., 2020). Deep learning approaches were recently used, achieving better results (Alharbi et al., 2021). We will explore three deep-learning models for ASR: Wav2Vec2, HUBERT, and Whisper.

Wav2Vec2 (Baevski et al., 2020) uses a self-supervised learning approach. They encoded the speech audio through a multi-layer CNN and then mask spans of the resulting latent speech representations. The architecture basically uses two modules: an encoder and a decoder. The encoder creates a numerical representation of the mel-spectrogram representation of the audio. A CNN network with 12 layers with a loss function is used to do this. The output is a matrix with a size of 1024 x 128. The decoder transforms the representation from the encoder into transcribed text, using an RNN network with 6 layers.

Hidden unit BERT (HuBERT) (Hsu et al., 2021) uses the same structure from Wav2Vec2 to process the input signal: a CNN encoder that generates representations from the audio mel-spectrograms, followed by a transformer encoder. The major difference is in the encoder layer, where the same strategy from BERT (Devlin et al., 2019) is used. Bidirectional Encoder Representations from Transformers (BERT) are based on the Transformers encoder-decoder architecture and self-attention mechanisms. BERT is pre-trained on a large corpus and can be fine-tuned for specific tasks (Devlin et al., 2019). For BERT, some words in sentences are masked, and then the model's objective is to predict the missing words. For HuBERT, the strategy is to use this on the Transformer hidden units, aiming to learn abstract representations of the speech.

The Whisper model was published in September 2022, and the architecture is based on an encoder-decoder Transformer. Whisper can process audio chunks within 30 seconds. The audio input is converted into a log-Mel spectrogram and sent to an encoder. A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and to-English speech translation (Radford et al., 2022).

2.3 Sentence Representation

Text representation techniques convert natural language text into a format that computers can process. In Natural Language Processing (NLP), we can represent different natural language elements, such as words, sub-words, sentences, or even entire paragraphs. Sentence embeddings represent sentences numerically through vectors in a high-dimensional space. This representation keeps the semantic relationship and makes extracting the meaning from a sentence possible. Since emotion recognition depends on the context to make sense, sentence embeddings can be a great alternative to this.

We can use specific models like the Sentence-BERT (SBERT) (Reimers and Gurevych, 2019). SBERT is a modification of BERT (Devlin et al., 2019), one of the state-of-the-art models for word embedding. Devlin et al. (2019) propose Bidirectional Encoder Representations from Transformers (BERT) based on the Transformers encoder-decoder architecture and self-attention mechanisms. BERT is pre-trained on a large corpus and can be fine-tuned for specific tasks. The results obtained by BERT achieved state-of-the-art for multiple NLP tasks, such as question answering, text classification, and named entity recognition. BERT is context-dependent, meaning the whole sentence is considered for the word embedding generation.

To deal with a fixed-size sentence embedding, considering BERT as input, SBERT uses a pooling operation at the end of BERT processing. This is necessary because BERT will generate an embedding array for each word in the sentence, and when dealing with different sentence sizes, each output will have a size. The mean calculation of all word embeddings was used as the default pooling operation.

Using these strategies, SBERT achieves state-of-the-art in some of the SentEval transfer tasks (Conneau and Kiela, 2018) that focus on sentiment prediction, such as on (1) MR, which focuses on sentiment in movie reviews, (2) CR, which focuses on customer product reviews, and (3) SST, the Stanford Sentiment Treebank. On MPQA, which focuses on opinion polarity, SBERT also achieves competitive results.

2.3.1 Models

Large pre-trained models can have some computational costs to execute. The MiniLM (Wang et al., 2020)

- Sentence BERT
  - mpnet
  - paraphrase-MiniLM-L3
  - all-MiniLM-L12

2.4 Recurrent Neural Network

Based on the human brain, statistics, and applied math, the Deep Learning (DL) process consists of learning from representations from the input data. The DL models can have multiple layers, extracting hierarchical features from the input data, making it possible for the network to focus on the most important ones for the task (Goodfellow et al., 2016). Deep Learning has been used to solve real-world problems in different domains,

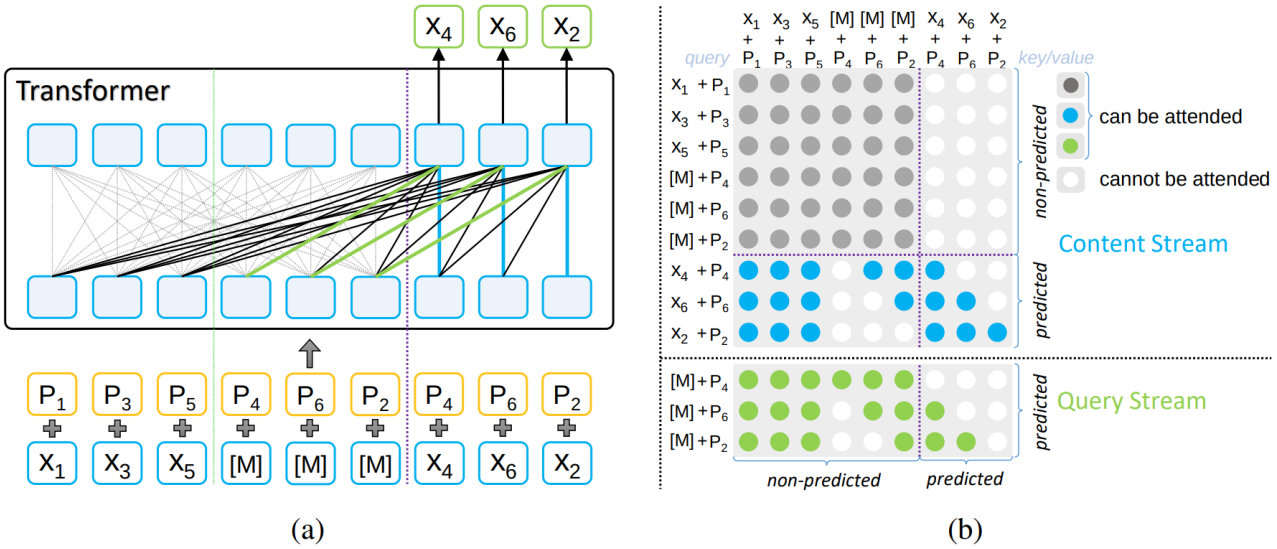

Figure 2.7 – MPNet architecture Song et al. (2020)

such as agriculture, psychology, health, and traffic, and can deal with different types of data input, like audio, video, image, and text. Combined types of data input are called multi-modal or cross-modal (Ngiam et al., 2011).

We have two different types of learning methods: supervised and unsupervised learning. In supervised learning, the model learns from labeled data. This means the model adapts to map input features into the correct labels throughout the training process. Once trained, the model can generalize to unlabeled data. On the other hand, unsupervised learning models can learn from unlabeled data. Therefore, these models can only deal with the input features and are often used to cluster similar elements or identify patterns based on their characteristics (Bhangale and Kothandaraman, 2022).

Focusing on speech emotion recognition tasks using bimodal data, the standard models use supervised learning, such as support vector machines, long short-term memory, convolutional neural networks, and, more recently, transformers (Geetha et al., 2024).

Long Short-Term Memory (LSTM) was proposed by Hochreiter and Schmidhuber (1997) and is a type of Recurrent Neural Network (RNN), which means it can keep long-term dependencies on sequential data. We detail the LSTM architecture in Figure 2.8. The architecture is composed of an input gate, which defines the information that will be added to the cell state; a forget gate that defines the information that will be removed from the cell state; an output gate, which defines the output from the LSTM; and a cell state that saves the information that passes through the LSTM. As it can use information from previous states to compute the result for new ones, LSTM is a good choice for audio signal processing.

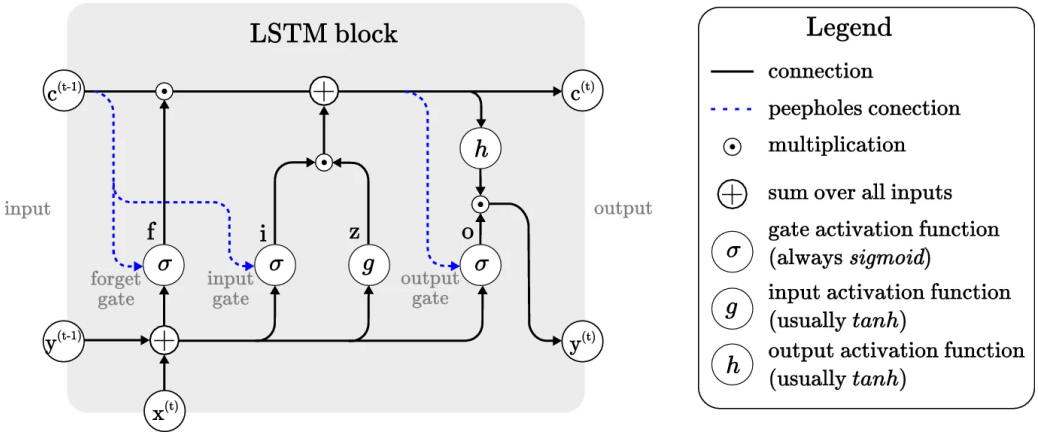

Figure 2.8 – Basic structure of LSTM Van Houdt et al. (2020)

2.5 Speech Emotion Recognition

The Speech Emotion Recognition (SER) task aims to recognize emotions from speech signals without the use of linguistic features Singh and Goel (2022). **?** summarize SER into two main parts: feature extraction and classification. In Figure **??**, we illustrate these two parts. As we can see, the Speech signal serves as input to the Front-End. In Front-End, we have the pre-processing and feature extraction steps that define a representation for the input. Such a representation is then fed to the Back-end, which has the ML classifier and the scoring function. As output to the Back-End, we obtain an Emotional state for the Speech signal.

Wani et al. (2021) introduce a six-step version for the process of speech signals. The first one is the preprocessing step, which is used for feature extraction. Features such as prosodic, spectral, voice quality, and MFCC are commonly extracted. In the second step, the signal framing divides the signal into smaller fixed-length sections, for example, 25 or 30 ms. This is necessary due to the limitation of input size in machine learning models, as well as to keep local features. In the third step, after the framing, windowing is used to treat the leakages that occur after the Fast Fourier Transform. A voice activity detector is used in the fourth step to get only speech parts from the utterance. Using this, unwanted noises and silence are removed from the signal. In the fifth step, normalization is used to standardize the sound volume. Finally, in the last step, noise reduction is used to remove unwanted noises present in speech.

There are many existing models applied for SER. In some cases, CNN models are used, treating SER as an image classification task, where the image of the Mel Spectogram feeds the model. On the other hand, models like SVM, decision trees, and autoencoder are used when using features such as prosodic, voice quality, and MFCC.

Datasets used to train and evaluate models have three types of origin: actor-based, induced, and natural emotion. Actor-based datasets are developed under labo-

ratory scenarios, where professional actors simulate emotions. While induced datasets consist of speakers exposed to stimuli that can bring specific emotions. Finally, Natural datasets contain emotions captured from speakers without any intervention or stimuli (Singh and Goel, 2022).

In general, existing approaches use two emotion classification methods to perform emotion recognition: discrete classes and dimensional evaluation. When using dimensional values, SER models are evaluated using the following metrics using Concordance Correlation Coefficient (CCC), in Equation 2.4.

CCC is the correlation between two variables that follow the Gaussian statistics, μ_1 and μ_2, and considering the standard deviations σ_1 and σ_2. The covariance is defined as σ_{12}.

$$CCC = \frac{2\sigma_{12}}{(\mu_1 - \mu_2)^2 + \sigma_1^2 + \sigma_2^2} \tag{2.4}$$

2.6 Data Streaming

Data streaming is a continuous flow of data that, by default, never ends (Andrade et al., 2014). Nowadays, we have a lot of streaming applications running: wearable sensors, traffic information, social media, streaming services such as Spotify, Netflix, autonomous cars, and many others applications. Data streaming has a data source or data producer, data tuples, and data schema in its structure. A data tuple is an atomic data item that an application will process in the data stream. Moreover, the data schema defines the structure of the data type in the tuple. Commonly, each tuple is associated with a timestamp (Andrade et al., 2014).

The data tuple can be structured, semi-structured, or unstructured. It is considered structured with a defined schema with name/type/values. A semi-structured data tuple does not have a defined schema, and, in some cases, it requires additional parsing and analysis to use it. Finally, unstructured data tuples consist of data that do not have patterns or are in a proprietary format (Andrade et al., 2014).

The process that will receive and process these tuples is called by Andrade et al. (2014) as data flow graphs. The operations that are applied to the incoming tuples in the data flow are classified as stateless or stateful. Stateless operations, as the name says, do not keep the state, and each tuple is processed without considering a previous history and the data arrival order. On the other hand, stateful operations involve pieces of information from other tuples and are more dependent on fault tolerance mechanisms.

Four operators are classified as stateless: projection, selection, aggregation, and split. The projection operator can add, remove, and update attributes of a tuple, which will produce a new tuple; the selection filters tuples. If the condition matches, the tuple will be selected. Otherwise, no; aggregation is similar to the group by function in SQL.

They make aggregations based on an attribute; split will divide the stream into multiple streams according to conditions determining which outbound streams will transport each tuple. Finally, stateful operators are sort, join, and barrier. Sort are windowed-based , which group and sort tuples based on a key value; join that are windowed-based and associate tuples based on a condition; barrier differs from join because they do not use match conditions. The barrier is also used to synchronize streams.

A data flow graph can be organized as a pipeline to build a streaming application. With a pipeline, it is possible to execute operations parallelly. We present an example in Figure 2.9 of a pipeline that performs two operations that generate an output sink. Sinks are defined as the consumers of the data produced by the streaming application, such as databases or files. In this example, each operation can run in parallel. Considering two input tuples, X and Y, after tuple X is processed in operator A, from the moment it starts processing in operator B, tuple Y can start to be processed by operator A.

Figure 2.9 – Pipeline example

Related to the data source, we can divide data streaming into two categories: event-based and continuous data. Event-based are triggered events that occur under certain conditions, for example, when someone starts talking. Furthermore, continuous data, as the name says, are in a constant flow, such as sensor data. As we have continuous incoming data, it is necessary to break it into smaller portions, making it possible to feed a deep learning model, for example. The process of doing this division is called windowing, where we have a segment/slice of stream ready to be processed. Akidau et al. (2018) defines three strategies for windowing: fixed, sliding, and session-based, as represented in Figure 2.10. Fixed windows are sliced with a fixed-size time-based length. Sliding windows divide data by a fixed length and period. Overlapping happens if the length is bigger than the period. When both are equal, windows are fixed. Finally, the window size can be dynamic, non-overlapping, and data-driven for session-based.

2.6.1 Flink

Further, stream processing is a natural paradigm for event-driven applications that need to react fast to real-world events and communicate with each other via message passing

- Kafka
- Flink

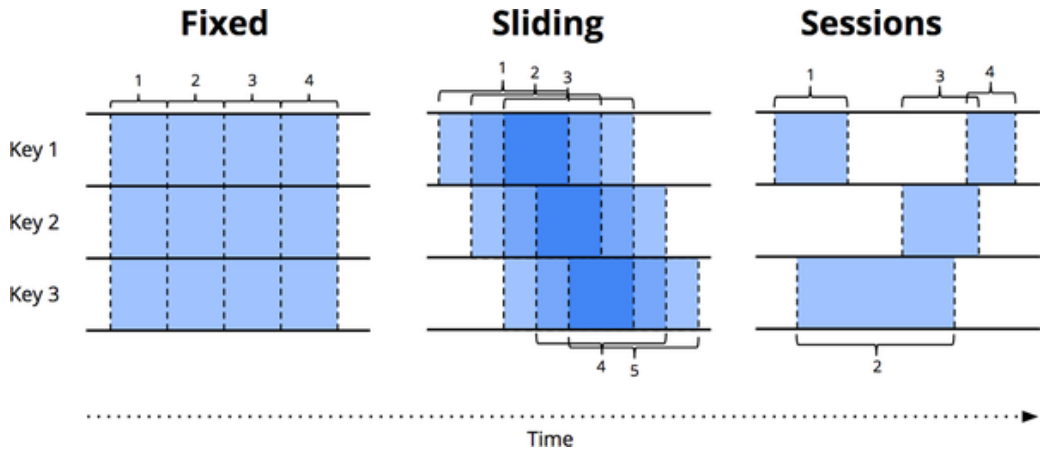

Figure 2.10 – Windowing strategies (Akidau et al., 2018)

3. RELATED WORK

In this section, we introduce some related works in the literature. For selecting them, we consider ten aspects: (1) the architecture used (how the approach processes the bimodal features); (2) if using a machine learning approach to extract acoustic features; (3) if using sentence embedding for text features; (4) dataset type: if is acted, or natural; (5) dataset language; (6) if it uses dimensions, what dimensions use, or (7) if it uses classes, how many classes; (8) if it uses streaming or not; (9) data type used; and (10) year of publication. For better understanding, we present a summary of all related works in Table 3.2.

We divide the analysis of the related works in two ways: the papers that use dimensional emotion recognition and text features, and the second analysis is related to papers that applied their models in a streaming scenario. This division was necessary because we did not identify any work that used a bimodal model with dimensional data in a streaming scenario. We have only a few models that use classes and audio-only data.

3.1 Dimensional Speech Emotion Recognition

Sun et al. (2020) introduce an approach that uses textual, acoustic, and visual information for dimensional emotion recognition. The proposed approach uses an LSTM with a self-attention mechanism and was trained and evaluated on the MuSe-CaR dataset. The use of the LSTM layer is due to the capability to get temporal dependencies. The authors also explore the use of different feature sets for each modality. Regarding textual information, they evaluated the following word embedding models: Glove, Word2Vec, and BERT. Using eGeMAPS, pAA, IS13, and VGGish were evaluated for acoustic features. Early and late fusion use were evaluated, and the best results were through late fusion. The early fusion involved concatenating features before feeding the network, while the late fusion used a second-level LSTM model that incorporated predictions from the unimodal features. When focusing on the use of textual and acoustic information, Sun et al. (2020) used IS13 with BERT and achieved 0.4931 of CCC for arousal, while for valence dimension, the authors used PyAudio with BERT-4 and achieved 0.4633 of CCC.

Atmaja and Akagi (2020) explored multitask learning for textual and acoustic features. The acoustic features evaluated were the LLDs and HSFs from GeMAPs and pAA. GloVe, FastText, and Word Embedding. Using three LSTM layers to process each modality individually, the authors use dense layers to concatenate the features. In that way, each input didn't need to have the same dimension. After the concatenation, the architecture has 2 dense layers with sizes 64 and 32, respectively, and the output is composed of three dense layers with size 1, representing each emotion dimension. The CCC is used

as loss, and calculated as $1 - CCC$. To train and evaluate the architecture, the IEMOCAP dataset was used, with a 7869:2170 split ratio. The annotation for arousal, valence, and dominance was normalized using a scale [-1,1]. The final approach used pAA HSF for acoustic features and WE + GloVe for text. In arousal, the CCC score was 0.571, the valence achieved 0.418 of CCC, and dominance with 0.500 of CCC. Besides evaluating the LSTM, Atmaja and Akagi (2020) also tested with the CNN network. However, the findings show that for multimodal dimension emotion recognition, the LSTM had better results.

Sogancioglu et al. (2020) investigate the use of TF-IDF, FastText, Polarity, FastText+Polarity, and Dictionary-based features for text features. The authors used a machine learning approach to extract information for acoustic features, using the Fisher Vector (Perronnin and Dance, 2007) as the encoder. The authors separate arousal and valence prediction according to the input features. They predicted arousal using acoustic features with a score-based decision fusion, while valence prediction was made with text features and a label-based decision fusion. The combination of Support Vector Machines (SVM) with Kernel Extreme Learning Machines (ELM) and Partial Least Squares (PLS) was used in the architecture. They evaluate their approach using the Ulm State of Mind Elderly (USOMS-e) dataset, obtaining an Unweighted Average Recall (UAR) of 63.7 for valence and 57.5 for arousal.

Focusing on early and late fusion models in an SVM model, Julião et al. (2020) uses BERT for textual features and the ComParE feature set combined with x-vectors. X-vectors are an audio embedding representation with 512 fixed dimensions. They evaluate the approach on the USOMS-e dataset using arousal and valence dimensions. The best results for arousal are 48.8% of UAR and 61% of UAR for valence. The results are through the early fusion, using the online and normalized version of x-vectors.

Atmaja and Akagi (2021) compare word embeddings, Word2Vec, and GloVe for textual features, and explore the GeMAPS feature set through LLDs, HSF1, and HSF2 configurations. The LLDs only use high-level statistical functions with a mean; HSF1 uses the mean and standard deviation of LLDs, and HSF2 uses the mean and standard deviation of LLDs and silence. The authors used an LSTM for each feature set and an SVM classifier to process the join of features. The evaluation was made using IEMOCAP and MSP-IMPROV datasets. To calculate how close the output values are to the gold standard, they use CCC. The best results were obtained through HSF2 combined with GloVe. On IEMOCAP, arousal achieves 0.579 of CCC, valence 0.553 of CCC, and dominance 0.465 of CCC. While on MSP-PODCAST, arousal gets 0.570 of CCC, valence 0.291 of CCC and dominance 0.405 of CCC.

Triantafyllopoulos et al. (2022) focuses on evaluating the impact of using a fine-tuned version of the w2v2-L-emo-ft model from Wagner et al. (2023) in the valence dimension, using the MSP-PODCAST dataset. The results for each dimension were arousal with 0.041, valence with 0.386, and dominance with 0.048 of CCC. With the experiments, the

authors confirm the hypothesis that the good results in valence from transformer-based models are due to the self-attention layers containing encoded linguistic knowledge.

Srinivasan et al. (2022) propose a teacher-student approach with a bimodal teacher model to fine-tune HuBERT. They train the teacher model as bimodal, using audio and text features, while the student model processes only audio embeddings. For textual features, the BERT pre-trained model was used. The authors evaluate the proposed approach on the MSP-Podcast and IEMOCAP. The CCC scores for the teacher model, which considers bimodal features, are on MSP-PODCAST: 0.765 for CCC in arousal, 0.690 for CCC in valence, and 0.683 for CCC in dominance. On IEMOCAP, the results are 0.668 for CCC in arousal, 0.648 for CCC in valence, and 0.537 for CCC in dominance.

Ispas et al. (2023) uses a multi-task and cross-attention architecture, where the output can be both categorical and dimensional emotion recognition. The HuBERT model was used to extract acoustic features, and for textual, the DeBERTaV3 was used. HuBERT and DeBERTaV3 have the same 1024 hidden dimension size; To maintain consistent dimensions, the shorter sequence is padded to match the longer one. The cross-attention involves the process of merging embeddings of the same dimension that originate from different modalities. IEMOCAP was used to train and evaluate the proposed approach. The CCC score for arousal was 0.677 and 0.748 for valence.

The use of text features, more precisely word embeddings, demonstrably improves results on the valence dimension. While the dominance and arousal are affected only by the acoustic features (Triantafyllopoulos et al., 2022; Srinivasan et al., 2022; Ghriss et al., 2022; Atmaja and Akagi, 2020, 2021; Sogancioglu et al., 2020; Julião et al., 2020). We notice the use of GloVe by (Atmaja and Akagi, 2020, 2021) and more recent approaches, such as BERT (Srinivasan et al., 2022; Julião et al., 2020; Sun et al., 2020) and a derivation of it called camemBERT (MacAry et al., 2021), and DeBERTaV3 (Ispas et al., 2023).

All of them use word representation level. We evaluate the use of sentence-level representations. This is because we will infer the emotion based on a sentence, not for each pronounced word. Keeping on that way, the meaning and the context of the words in the sentence.

In this set of papers, we found a focus on the acoustic features used. For example, we used eGEMAPS and ComParE feature sets, which improved SER results. Considering emotions, audio embeddings were explored in the music emotion recognition task. Koh and Dubnov (2021) evaluate L3-Net (Cramer et al., 2019) and VGGish models. For SER, Julião et al. (2020) explored the use of x-vectors (Snyder et al., 2018) embedding, Sun et al. (2020) even evaluated the use of VGGIsh, but not for the bimodal approach. More recent approaches consider the use of w2v2 (Triantafyllopoulos et al., 2022) and HuBERT (Ispas et al., 2023; Srinivasan et al., 2022) to generate the representations. Independent of the method to extract the features from the audio, even using pre-trained models or hand-crafted options, none of them had the time necessary to process this in-

formation. Our approach compares the ComParE, eGeMAPS, pAA feature sets, and TRILL and VGGISH models for audio embeddings.

3.2 Speech Emotion Recognition in Streaming Environment

We find three different approaches for speech emotion recognition that run in a streaming environment. Bertero et al. (2016) built a dataset from the TED-LIUM corpus and used six categories of emotion: criticism, anxiety, anger, loneliness, happiness, and sadness. To make it possible to use in real-time, their approach uses a CNN model and the raw audio as input, down-sampled at 8 kHz. The accuracy for each class was Criticism/Cynicism 61.2%, Defensiveness/Anxiety 62.0%, Hostility/Anger 72.9%, Loneliness/Unfulfillment 66.6%, Love/ Happiness 60.1%, Sadness/Sorrow 71.4%. To classify, the time necessary to process each second of speech was 13 ms.

Stolar et al. (2017) uses a different approach, considering speech recognition as an image classification task. They used the spectrogram image to feed the model to make this possible. The authors evaluated their approach using the Berlin Emotional Speech (EMO-DB) dataset. Two different approaches were tested; FTAlexNet achieves better accuracy, while the AlexNet-SVM uses fewer computations. The average accuracy with the FTAlexNet model for female voices was 79.68%, and 76.79% for male voices.

Lech et al. (2020) focus on evaluating the impact of reducing the speech bandwidth for SER, using categories. Seven emotions were considered: anger, happiness, sadness, fear, disgust, boredom, and neutral speech. The CNN model was used to realize the predictions. With CNN, the spectrogram was used to feed the model. The approach was trained and evaluated on Berlin Emotional Speech (EMO-DB). In a real-time environment, the prediction is done every 1.033–1.026s. The baseline accuracy on EMO-DB was 82%, and the reduction of bandwidth from 8 to 4 kHz decreased by an accuracy of 3.3%.

Unlike these approaches, we will use dimensional emotion recognition instead of discrete classes. Also, our focus is on bimodal features, while Stolar et al. (2017), Bertero et al. (2016) and Lech et al. (2020) use only acoustic features. Another point is that these papers are from before 2020, and after that, we do not have publications that focus on SER that run on a streaming environment, different from the ASR task, where we have some new approaches over the years, such as Dominguez-Morales et al. (2018); Singh et al. (2019); Leow et al. (2020) and Saeki et al. (2021). It is important to notice that only Lech et al. (2020) provides metrics for evaluating real-time / streaming scenarios. Stolar et al. (2017) and Bertero et al. (2016) only mentioned that their approaches are in real-time but do not show the result.

Ref	Architecture	Audio Embedding	Sentence Embedding	Dataset Type	Language	Dimensions Evaluated	Total Classes	Streaming	Data Type	Year
Bertero et al. (2016)	CNN	No	No	Natural	English	-	6	Yes	Audio	2016
Stolar et al. (2017)	FTAlexNet	No	No	Acted	German	-	7	Yes	Audio	2017
Lech et al. (2020)	CNN	No	No	Acted	German	-	7	Yes	Audio	2020
Sun et al. (2020)	Self-Attention + LSTM	No	No	Natural	English	AVD	-	No	Audio Text	2020
Atmaja and Akagi (2020)	LSTM	No	No	Acted	English	AVD	-	No	Audio Text	2020
Sogancioglu et al. (2020)	SVM	No	No	Natural	German	AV	-	No	Audio Text	2020
Julião et al. (2020)	SVM	Yes	No	Natural	German	AV	-	No	Audio Text	2020
Atmaja and Akagi (2021)	SVM	No	No	Acted Natural	English	AVD	-	No	Audio Text	2021
Triantafyllopoulos et al. (2022)	w2v2 fine-tuning	Yes	No	Natural	English	AVD	-	No	Audio Text	2022
Srinivasan et al. (2022)	Conditional Teacher-Student	No	No	Natural Acted	English	AVD	-	No	Audio Text	2022
Ghriss et al. (2022)	LSTM	No	No	Natural	English	AVD	-	No	Audio Text	2022
Triantafyllopoulos et al. (2023)	MFCNN14	Yes	No	Natural Acted	English	AVD	-	No	Audio Text	2023
Ispas et al. (2023)	Transformer	Yes	No	Natural	English	AVD	-	No	Audio Text	2023
Our Approach	**LSTM**	**Yes**	**Yes**	**Acted**	**English**	**AVD**	**-**	**Yes**	**Audio Text**	**2024**

Table 3.1 – Related Works

4. SPEECH EMOTION RECOGNITION ON STREAMING

Dimensional Speech Emotion Recognition has many potential applications in the real world. Using dimensions, it is possible to map and identify anxious traces and reactions, check if a class is boring to the students, detect if a driver is tired while driving, determine the level of satisfaction of customers, among others. However, there is a gap between the literature and the real world, where we have many approaches for SER, but no one is built to support real-world scenarios where we have real-time information incoming. Models that run on a streaming environment must be fast enough to bring results as soon as information arrives, but they also need good output accuracy. Because of this, this work aims to unify SER, deep learning, and streaming to build a robust approach that can be applied to the real world.

4.1 Datasets

To evaluate our experiments, we use the IEMOCAP (The Interactive Emotional Dyadic Motion Capture) dataset (Busso et al., 2008). IEMOCAP contains multimodal information, combining video, speech, motion capture of face, and text transcriptions. From these features, we only use speech and text transcriptions. In total, the dataset contains approximately 12 hours of speech. IEMOCAP provides a VAD score and an emotion class annotation for each utterance. VAD scores range from 1 to 5. The dataset contains approximately 12 hours of speech.

Since IEMOCAP does not contain information about the split ratio, we divided it into 60/20/20 ratios for training, testing, and validation. The validation set was used to compute the results of all experiments. In total, the 1992 utterances from the dataset have 8909 seconds of duration.

The second dataset is the MSP-PODCAST (Lotfian and Busso, 2019). MSP-PODCAST is a natural-based dataset built with different podcast records. In total, there are 237 hours and 56 minutes of annotated utterances. The annotation values for dimensions use a scale of 1 to 7 for valence, arousal and dominance. The categorical labels complain: anger, happiness, sadness, disgust, surprise, fear, contempt, neutral and other. In contrast to IEMOCAP, MPS-PODCAST provides split information for Train, Development, Test1, and Test2. The train set contains 84,030 segments, the development set 19,815 segments, the test1 30,647 segments and the test2 14,815 segments.

We normalized to a -1 to 1 scale with the Equation 4.1. This normalization is since the original Russel approach uses the -1 to 1 scale, which will be the pattern we will use in our final architecture.

$$\frac{x - \left(\frac{max - min}{2} + 1\right)}{\frac{max - min}{2}} \tag{4.1}$$

4.2 End-to-End Speech Emotion Recognition Architecture

Our end-to-end architecture is composed of two blocs. The front-end and the back-end. The front-end is responsible for extracting the features from the input signal, while the back-end is responsible for processing the information from the front-end and predicting the output. The architecture is detailed in Figure 4.1.

After receiving the raw audio, we transform it into a mono waveform and resample it into a 16 kHz sample rate if necessary. The limitation length is 10s of audio due to the VGGIsh limitation. We extract two levels of features from the waveform: textual and acoustic.

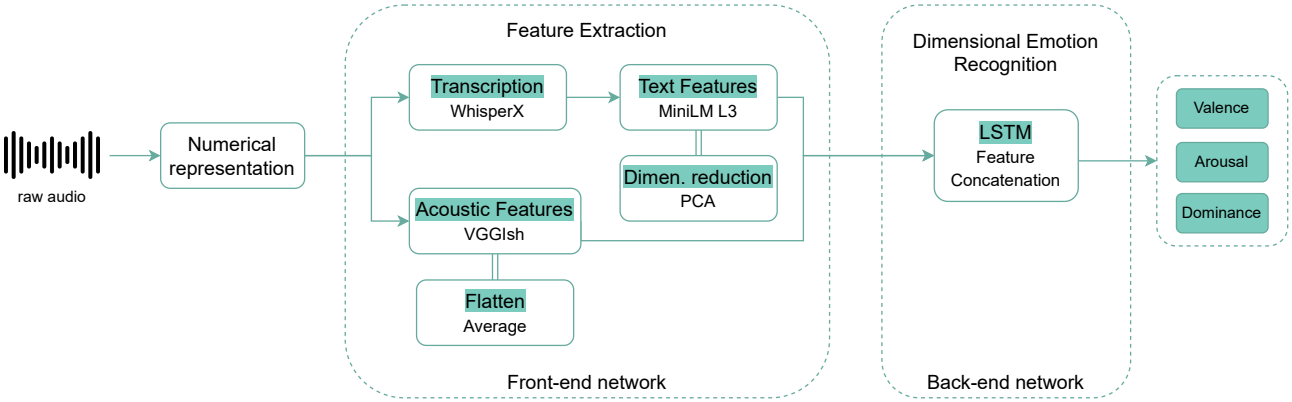

Figure 4.1 – End-to-End Speech Emotion Recognition Architecture

The objective of the front-end is to extract and pre-process the textual and acoustic features, providing the correct shape to the back-end network concatenate and process it. The expected output is two vectors with 128 dimensions each. For acoustic features, we generate audio embedding using the pre-treined VGGIsh model. VGGIsh generates a vector with 128 dimensions for each second of audio. We use the average from each element in the matrix as a flattened function, generating a unique vector for our back-end network.

The text features require an extra processing stage. We use the WhisperX model to convert the input waveform into text, thus allowing the generation of sentence embedding for textual representation. To generate the sentence embedding, we use the MiniLM L3 pre-trained model that generates a vector with 384 dimensions. We apply the dimensionality reduction with the Principal Component Analysis (PCA) algorithm to reduce the dimensions to 128.

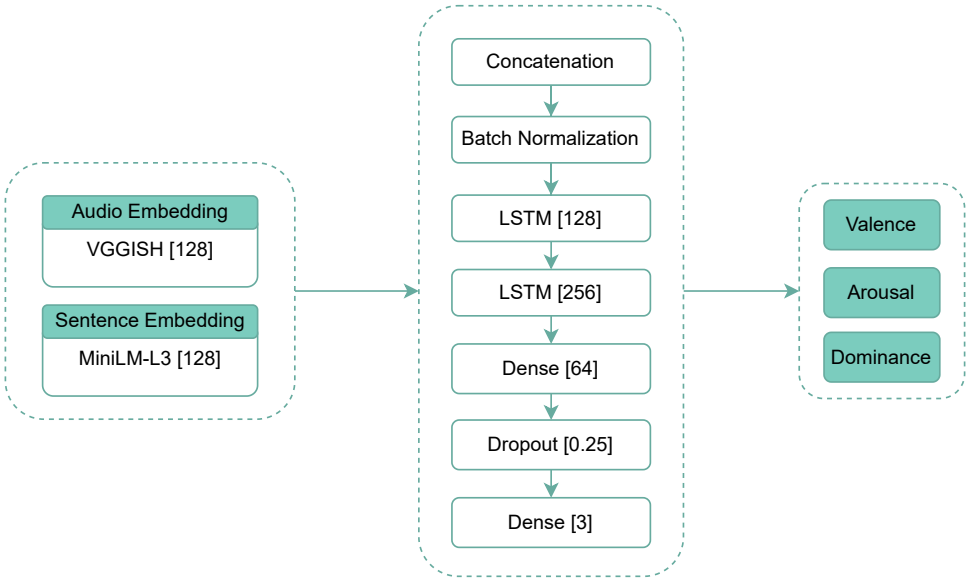

Figure 4.2 – Back-end Architecture

The back-end network uses an LSTM network to process the incoming data. The first layer concatenates both feature sets. We use the order *audio*, *text*. After the input layer, we use a batch normalization layer to standardize the features. We use only two LSTM layers, the first with 128 units, and the second with 256 units, followed by a dense layer with 64. We apply a dropout with a 0.25 probability. The output is a dense layer with 3 values corresponding to valence, arousal, and dominance dimensions. We use tahn as the activation function and Adam optimizer with a 0.001 learning rate.

4.3 Evaluation Results

We train and evaluate our model on two open datasets: IEMOCAP and MSP-PODCAST. We further detail on Section 4.1. On IEMOCAP, we used the solution provided by Atmaja and Akagi (2020) as a baseline to compare our approach. As detailed in Chapter 3, Atmaja and Akagi (2020) also uses an LSTM model with GloVe for textual features combined with pAA HSF for acoustic features.

The main point in defining our architecture is the time necessary to process the incoming data. While Atmaja and Akagi (2020) focuses on word embedding, with GloVe, we focus on capturing the sentence's meaning through the sentence embedding from MiniLM L3. The MiniLM L3 was tested on the Sentiment Analysis task and performed well on Stanford Sentiment Treebank (SST) Socher et al. (2013). The textual embedding focuses on improving the valence dimension; the task is close to sentiment analysis, going from negative to positive perspectives.

On the acoustic side, the use of VGGIsh to recognize emotions has been explored by Pham et al. (2023) in bimodal categorical speech emotion recognition and by Koh and

Mode	**CCC**		
	Valence	**Arousal**	**Dominance**
Baseline			
Bimodal LSTM (GloVe + HSF from pAA) (Atmaja and Akagi, 2020)	**0.418**	0.571	0.500
Our approach			
VAD MiniLM-L3 VAD	0.4165	0.2989	0.2989
LSTM Concat (VGGISH + MiniLM-L3 PCA) VAD	0.1431	**0.5915**	**0.5899**

Table 4.1 – IEMOCAP evaluation results

Dubnov (2021) in music emotion recognition. Pham et al. (2023) uses the concatenation of VGGIsh and BERT to recognize emotions. In addition to the mode to recognize emotion, the main difference in our approach is in the architecture used and the textual representation. Originally, VGGIsh was trained to focus on audio classification tasks and achieved better results than hand-crafted features on the Audio Set Acoustic Event Detection (AED) classification task. Using GPU, the processing time of VGGIsh took 0.0133ms per second of audio, while the approach of Atmaja and Akagi (2020) uses pAA with 9.9ms per second (see Table 5.1.1).

5. EXPERIMENTS

We detailed the process to define the architecture in Figure 5.1. The first step is to select the best way to represent the textual and acoustic information. The second one is important to determine the best way to use both representations in our model. We will discuss each step in the next section.

5.1 Feature Selection

To perform SER in a streaming scenario, choosing optimal libraries or models to generate the representations of the input sources is necessary. We aim to explore the use of textual and acoustic information. To make this possible, we define a set of experiments to select the optimal choice for (1) transcribing the audio, (2) generating representation for acoustic information, and (3) generating sentence embeddings for textual information. This set of experiments is detailed in Figure 5.1-[A].

To evaluate, we use the IEMOCAP dataset. Experiments 1 and 3 use the full dataset, while experiment 2 uses only the test set. The objective for each one is to define the best option considering the speed to process the data and a lower error rate on evaluation.

We consider only pre-trained models for (1) automatic speech recognition task. To make the transcription, we select the state-of-the-art models and compare the speed (time used to transcribe a chunk of audio) and the Word Error Rate (WER) (Equation 2.3) to measure the quality of the transcribed text. In our tests, we evaluate: Wav2Vec2 (Baevski et al., 2020), WhisperX (Bain et al., 2023) (using the whisper v2 large as base model), fine-tuned XLSR-53 Wav2Vec2 (Grosman, 2021), HuBERT (Hsu et al., 2021), Seamless M4T v2, and Whisper v3.

Considering the different ways to (2) generate a representation for acoustic information, we evaluate two approaches: handcrafted features and audio embeddings. We use OpenSmile and pAA libraries for eGeMAPS, ComParE, and pAA sets to extract handcrafted features. OpenSmile library permits the extraction of two different levels of information: the low-level descriptors and the functionals. We compare both versions for eGeMAPS and ComParE. To generate audio embedding, we use the pre-trained VGGish and TRILL models.

Since we aim to keep the sentence's meaning for recognizing emotion, we define the use of (3) sentence embeddings for textual information. SBERT model has a good performance on sentiment classification, so we consider the following models from Sentence

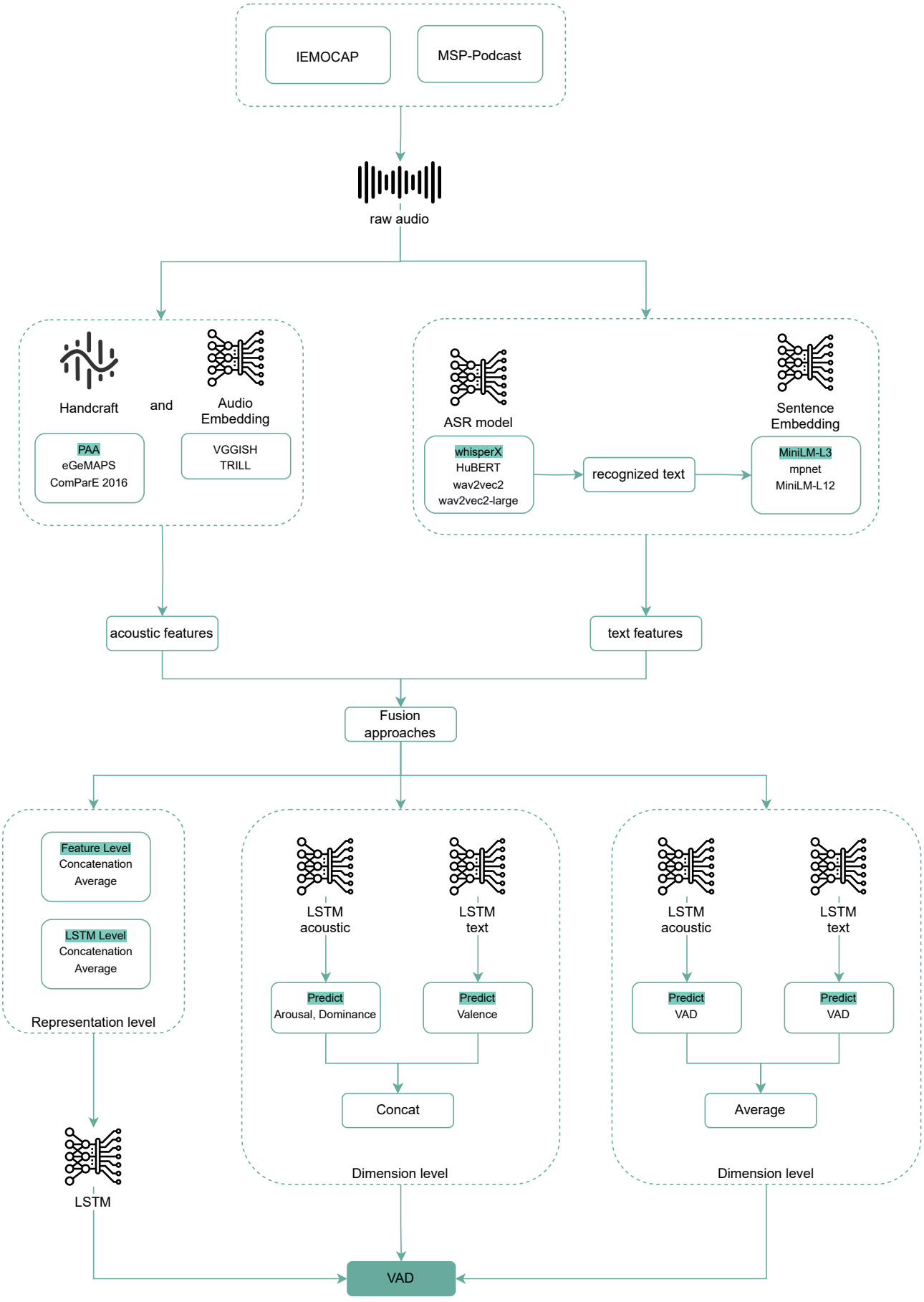

Figure 5.1 – The complete process for speech emotion recognition framework

Transformer library [1] to generate the embeddings: MiniLM-L12, mpnet, and MiniLM-L3. We selected them based on the speed reported on the documentation.

Experiments (2) and (3) are evaluated through an LSTM network to predict valence, arousal and dominance. LSTM is a good choice for work with sequential data, and the architecture of our network was based on a previous work from Atmaja and Akagi (2021) (check ref). We detail the architecture in Figure 5.2.

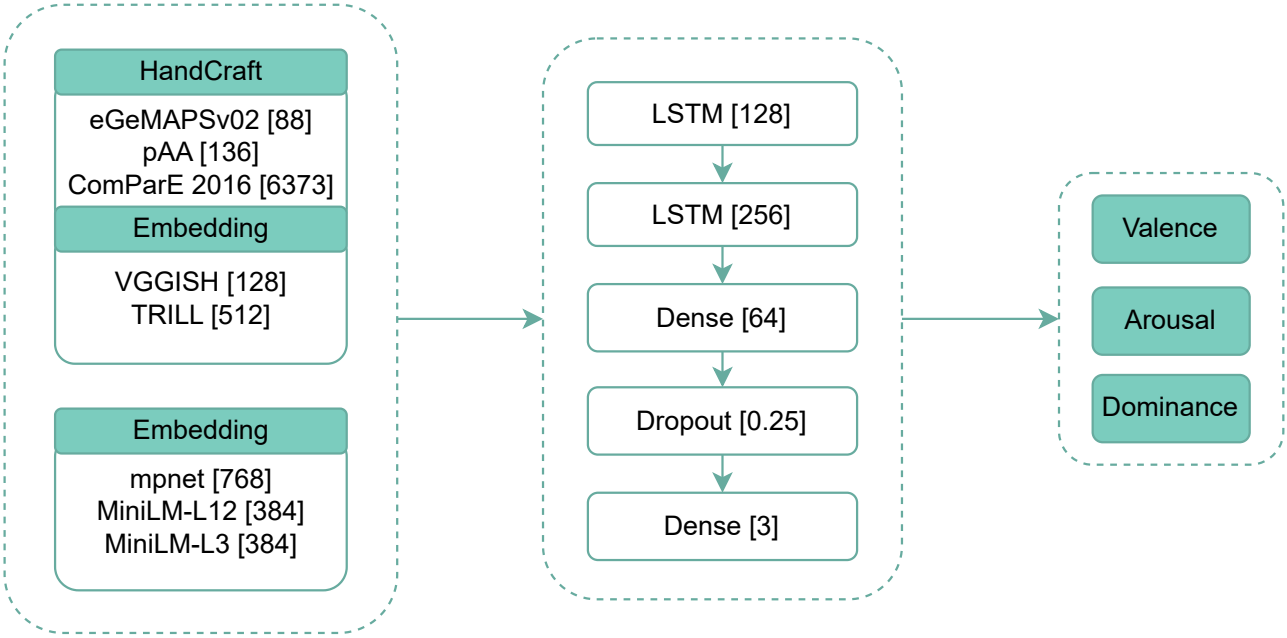

Figure 5.2 – LSTM architecture for acoustic and text features

We used the pAA feature set as a base to define the LSTM architecture for evaluating all the other features. This is necessary because we aim to use the same architecture for all the feature sets. We used a script that used all possible combinations for the parameters in Table 5.1.

Parameter	**Value**
Dropout	0.5, 0.25
Learning Rate	0.1, 0.01, 0.001
Optimizer	SGD, ADAM, RMSPROP
Batch Size	32, 64, 128, 256
Epochs	10, 50
Activation Function on LSTM	linear, tanh
Output Activation Function	linear, tanh

Table 5.1 – LSTM experimental configuration set

Our LSTM architecture was implemented using the Keras framework [2]. The final parametrization, considering all the possibilities based on the parameters in Table 5.1,

[1] https://www.sbert.net/
[2] https://keras.io/

consists of two LSTM layers, the first with 128 units followed by one with 256 units. Tanh is used as the activation function; one dense layer with 64 units is followed by a dropout layer with a probability of 0.25, and finally, the output is a dense layer with three dimensions. Both networks use Adam optimizer and 0.01 as the learning rate. The training uses batch sizes of 256 and 100 epochs, and loss calculation uses Mean Squared Error (MSE) (Equation 5.1). The input size is based on the feature dimension, represented in the first column of Figure 5.2.

$$\sum_{i=1}^{D}(x_i - y_i)^2 \tag{5.1}$$

The pre-processing consists of extracting and storing the features into Numpy files, using the split into three sets: train, develop, and test. For this process, we evaluate the time necessary to generate the whole dataset using each representation option. After that, we load these files and feed the LSTM network. Before feeding the LSTM, we use the *StandardScaler* function from sklearn [3] preprocessing to standardize the features. The standard calculation for a x feature is represented in Equation 5.2. Where we have the subtraction of the mean and the division by the standard deviation. In that way, adjusting the distribution of the feature. The evaluation of the MSE and CCC was made through the prediction function from Keras.

$$z = (x - \mu)/s \tag{5.2}$$

5.1.1 Results

In this section, we will discuss the findings of each experiment. In automatic speech recognition, the lower WER was achieved by the Whisper v3 model, with 0.2262. Table 2.3 shows the complete results for each model. Wav2Vec2 had a better performance, with 1.33s; however, considering WER, it has a big difference compared to Whisper v3, where Wav2Vec2 achieves 0.9881. Thinking about the better choice for our scenario, the best option is WhisperX, considering the second-lowest WER, with 0.2738, and the second-highest processing time, with 2.803s.

We achieved distinct results for experiments (2) generating representation for acoustic information and (3) generating sentence embeddings for textual information. The complete result is detailed in Table 5.1.1. Although there is a broad use of handcrafted features in the literature, the processing time to extract the features is relatively high compared to an audio embedding model, like VGGIsh. pAA has the second faster time, with 81.357s, while the best option is VGGIsh, with 26.47s. The main focus for acoustic features is the arousal and dominance dimensions that have more impact on acoustic

[3] https://scikit-learn.org/

Model	Time (GPU)	WER
HuBERT	3.2162	0.9643
Wav2Vec2	***1.3373***	0.9881
WhisperX	2.803	0.2738
Whisper v3	65.6492	***0.2262***
Wav2vec2 Large xlsr	3.2454	0.5595
Seamless M4T v2	23.0189	1.0238

Table 5.2 – Automatic Speech Recognition Evaluation

information. Even with eGeMAPS getting better results on CCC, the processing time is too high to be used in a streaming scenario. So, in this case, the best option is VGGIsh, which has a lower processing time and a competitive CCC compared to pAA and eGeMAPS.

Input	CCC/MSE V	CCC/MSE A	CCC/MSE D	Time(s)	thrg.(ms)
Acoustic Evaluation					
ComParE LLD	0.025 / 0.2045	0.1196 / 0.114	0.1156 / 0.1139	483.8202	40.27
ComParE	0.1978 / 0.1887	0.5308 / 0.0769	0.5308 / 0.0769	526.5632	46.88
eGeMAPS LLD	0.0108 / 0.2045	0.0792 / 0.1159	0.0789 / 0.1159	509.2656	59.85
eGeMAPS	0.2052 / 0.1819	0.6066 / 0.0704	0.6086 / 0.0709	509.2656	59.85
pAA	0.136 / 0.1923	0.5813 / 0.075	0.5803 / 0.075	81.357	9.9
TRILL	0.1978 / 0.1887	0.5308 / 0.0769	0.5308 / 0.0769	1099.2904	160.09
VGGISH	0.1751 / 0.1932	0.5694 / 0.0751	0.5694 / 0.0751	**26.4742**	0.0133
Text Evaluation					
MiniLM-L12	0.1292 / 0.1952	0.0917 / 0.1192	0.0913 / 0.1192	11.9568	
mpnet	0.0412 / 0.2022	0.0245 / 0.1202	0.0226 / 0.1193	11.8198	
MiniLM-L3	0.3238 / 0.1875	0.2057 / 0.1125	0.2057 / 0.1125	4.1988	

Table 5.3 – Acoustic features results

In experiment (2), the processing is around 6 times faster than the one for audio. As we can see in Table 5.1.1, MiniLM-L3 is the faster model for sentence embedding generation, with a 4.2s. Considering the CCC, MiniLM-L3 also achieved the highest value for valence, with 0.3238. In this case, MiniLM-L3 is the best option in both evaluation cases.

Using this LSTM architecture, the expected processing time for each second of audio input is 0,78ms for transcribing and generating sentence embedding and 2,97ms for generating the audio embedding. In the next section, we will present some ablation studies to get better results using these representations.

5.2 Fusion Approaches

Once the features used to represent the acoustic and textual data are defined, we evaluate the best way to use both types of information. To do this, we followed some of the approaches reviewed by Atmaja et al. (2022). We consider the (1) model level,

(2) feature level, (3) decision-level fusion, and (4) average from acoustic and linguistic features.

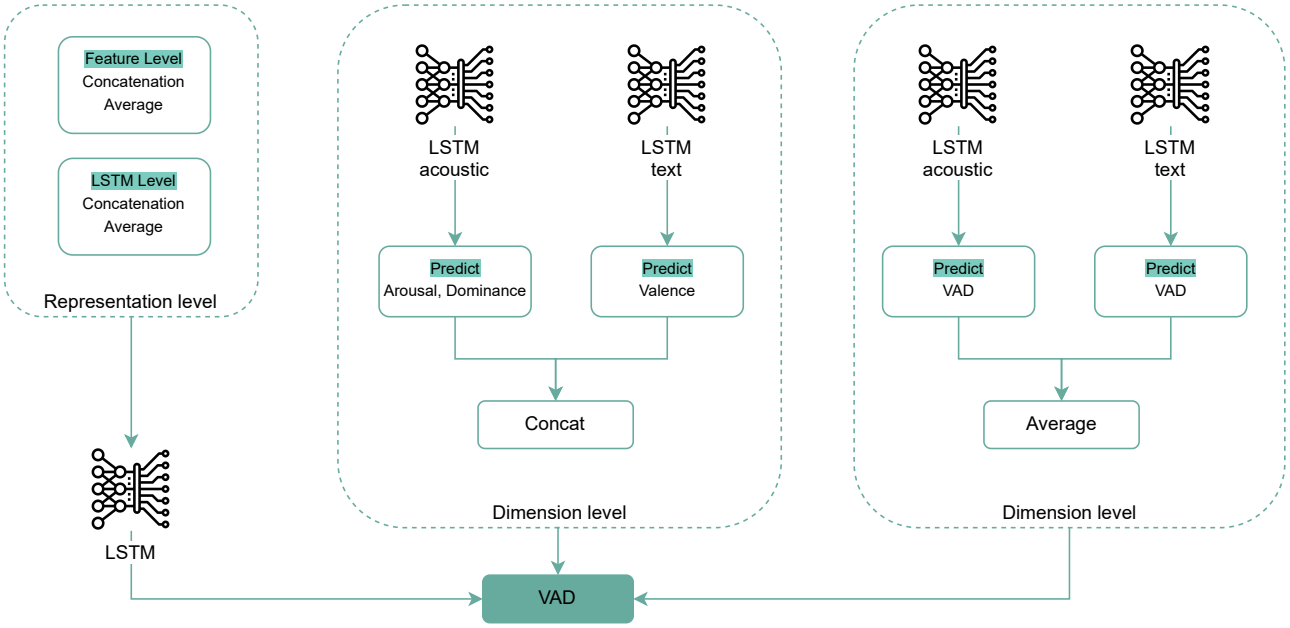

Figure 5.3 – Different structures for fusion concatenation

We detail our experiments in Figure 5.3. At the representation level, we have four approaches, considering average and concatenation of (1) model and (2) feature level. Considering the model approach, we used an extra keras layer before the batch normalization layer. For both cases, we first use the audio embedding and, in sequence, the sentence embedding data. Using the new keras layer, reducing the dimensionality of the sentence embedding was necessary to be equal to the VGGIsh dimension size. To do this, the PCA matrix decomposition from sklearn was applied to reduce from 384 to 128. At the feature level, we used the Numpy Concat function for the concatenation of both features and for average, we first applied the same PCA function; after that, we used the Numpy Average function.

Considering the dimension level, we used two approaches: (3) decision-level fusion and (4) average from acoustic and linguistic features; we used two LSTM networks to process the acoustic and textual information. In the first case, we trained using the audio embedding an LSTM with two outputs: arousal and dominance dimensions, while on the other hand, we trained an LSTM with the sentence embedding only for the valence dimension. The other approach uses the three dimensions and the same structure for the LSTM; we only calculate the prediction average for audio and sentence embeddings.

5.2.1 Results

We verified that optimizing some parameters of our LSTM brings better results for unimodal approaches. But this is not our focus here. An interesting behaviour is that using fewer dimensions on output significantly worsens the results. Using the average on Valence, the CCC is lower than 0.1

Mode	**CCC/MSE**		
	Valence	**Arousal**	**Dominance**
Dimension Level			
VAD VGGISH VAD	0.1482 / 0.1986	0.5533 / 0.0725	0.5528 / 0.0724
VAD MiniLM-L3 VAD	**0.4165** / 0.1954	0.2989 / 0.1223	0.2989 / 0.1223
VAD MiniLM-L3 PCA VAD	0.1055 / 0.2725	0.0805 / 0.143	0.0805 / 0.143
V Avg (MiniLM-L3 V + VGGISH V) **AD** ComParE AD	0.0186 / 0.2214	0.0996 / 0.1145	0.0981 / 0.1146
V Avg (MiniLM-L3 3 VAD + ComParE VAD) **AD** ComParE VAD	0.0852 / 0.1933	0.1171 / 0.1103	0.1156 / 0.1104
Representation Level - Manual Concatenation			
VGGISH + MiniLM-L3 VAD	0.4034 / 0.1977	0.2883 / 0.1317	0.2883 / 0.1317
Representation Level - LSTM			
Concat (VGGISH + MiniLM-L3 PCA) VAD	0.1431 / 0.203	**0.5915** / 0.0725	**0.5899** / 0.0725
Average (VGGISH + MiniLM-L3 PCA) VAD	0.0555 / 0.2007	0.434 / 0.0872	0.4325 / 0.0873

Table 5.4 – Fusion evaluation results

Working on the representation level, using the manual concatenation before passing to the LSTM input layer, we achieved better results than the dimension level, but it does not make sense to use when compared with unimodal features. Valence is lower than with only MiniLM, while arousal and dominance are lower than with VGGIsh.

The results were interesting when using the concatenation at the LSTM level as a Keras layer. We have increased arousal and dominance CCC scores, achieving 0.5915 and 0.5899, respectively. We also tested the order of features in concatenation, and the best option is to use VGGIsh first. (Bring results?) The average layer has lower results than only VGGIsh features.

Based on this, our final approach will use the concatenation of VGGIsh and MiniLM-L3 with PCA.

5.3 Streaming

The streaming implementation took place in two ways: one for evaluation and the other for real-world application. This is necessary since there are no datasets available for streaming scenarios. So, to make the evaluation possible, we iterate over the data,

preserving the duration of each file annotated. In the real-world scenario, we used a window time-based to split the incoming signal. We present the architecture in Figure 5.4.

To generate the audio input streaming, we use the pyAudio streaming function to capture the signal from the microphone as mono. We specify the params used to capture the audio in the Table 5.5. The number of chunks is calculated by multiplying the chunk length and the sample rate. The chunk represents the number of frames into a mel spectrogram input, calculated over the number of samples divided by the hop length. We use a mono channel.

Parameter	Value
Sample Rate	16000
N FFT	400
N MELS	80
Hop Length	160
Chunk Length	30
Number of Samples	CHUNK LENGTH * SAMPLE RATE
Chunk	N SAMPLES / HOP LENGTH
Format	pyaudio.paInt16
Channels	1

Table 5.5 – pyAudio parameters for audio capturing

After the windowing process, we convert the input signal into a numerical representation. We use the Whisper function, which uses FFmpeg to convert the signal into a waveform. After that, we use the Kafka producer to send the waveform to the queue, which Flink will process. To predict the values for valence, arousal, and dominance, we created an API using Flask to receive the requests from Flink. We use an API because Tensorflow models cannot be used in a streaming environment. We also make tests with Spark Streaming, but it only works using batches, which is not our objective.

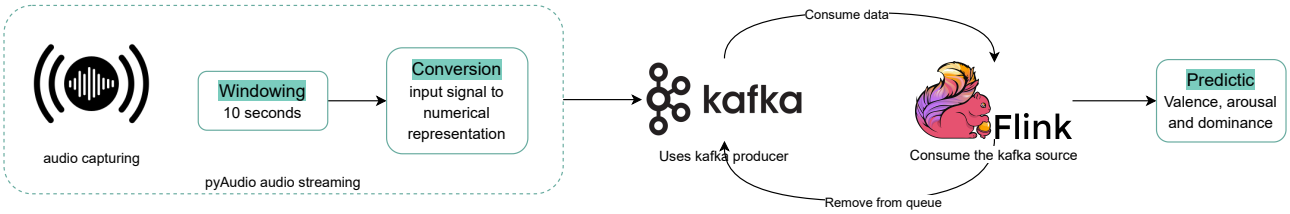

Figure 5.4 – Architecture used for streaming speech emotion recognition

Our API has four different endpoints; in that way, we can use different producers in Flink. First, we transcribe and generate the audio embedding. After that, using the transcription, we generate the sentence embedding and apply the PCA to reduce dimensionality. With both embeddings, we predict the three dimensions using our LSTM model. After getting the prediction, we remove the waveform from the Kafka queue.

5.4 Discussion

- PCA is bad
- LSTM is the better option?
- ML options for acoustic features: wav2vec2, w2v bert 2.0, hubert and so on
- Why meaning of sentence
- Fast and furious
- Application scenarios
- Etics

5.5 Reproducibility

We perform our experiments on our laboratory server, which has the following specifications: Operating system:

- Ubuntu 20.04.4 LTS
- Kernel: Linux 5.4.0-109-generic
- Architecture: x86-64

Hardware specification:

- CPU: AMD Ryzen 5 5600X 6-Core Processor with 12 threads
- Memory: total memory space 32058 (MB)
- GPU: NVIDIA GeForce RTX 3090 (24576 MiB)

6. CONCLUSION

Future work Pérez-Toro et al. (2022)

6.1 Future Work

REFERENCES

Ahn, J., Gobron, S., Silvestre, Q., and Thalmann, D. (2010). Asymmetrical facial expressions based on an advanced interpretation of two-dimensional russells emotional model. In: *proceedings of ENGAGE 2010*.

Akidau, T., Chernyak, S., and Lax, R. (2018). *Streaming systems: the what, where, when, and how of large-scale data processing*. O'Reilly, Sebastopol, CA, first edition ed.. OCLC: ocn975362965.

Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., Alharbi, S., Alturki, S., Alshehri, F., and Almojil, M. (2021). Automatic speech recognition: Systematic literature review. *IEEE Access*, vol. 9, pp. 131858–131876.

Andrade, H. C. M., Gedik, B., and Turaga, D. S. (2014). *Fundamentals of Stream Processing: Application Design, Systems, and Analytics*. Cambridge University Press, USA, 1st ed..

Atmaja, B. and Akagi, M. (2020). Dimensional speech emotion recognition from speech features and word embeddings by using multitask learning. *APSIPA Transactions on Signal and Information Processing*, vol. 9.

Atmaja, B. and Akagi, M. (2021). Two-stage dimensional emotion recognition by fusing predictions of acoustic and text networks using svm. *Speech Communication*, vol. 126, pp. 9–21.

Atmaja, B. T., Sasou, A., and Akagi, M. (2022). Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion. *Speech Communication*, vol. 140, pp. 11–28.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, vol. abs/2006.11477.

Bain, M., Huh, J., Han, T., and Zisserman, A. (2023). Whisperx: Time-accurate speech transcription of long-form audio. *INTERSPEECH 2023*.

Bertero, D., Siddique, F. B., Wu, C.-S., Wan, Y., Chan, R. H. Y., and Fung, P. (2016). Real-time speech emotion and sentiment recognition for interactive dialogue systems. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1042–1047, Austin, Texas. Association for Computational Linguistics.

Bhangale, K. B. and Kothandaraman, M. (2022). Survey of Deep Learning Paradigms for Speech Processing. *Wireless Personal Communications*, vol. 125, pp. 1913–1949.

Boehner, K., DePaula, R., Dourish, P., and Sengers, P. (2005). Affect: From information to interaction. In: *Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility*, CC '05, pp. 59–68, New York, NY, USA. Association for Computing Machinery.

Brunet, K., Taam, K., Cherrier, E., Faye, N., and Rosenberger, C. (2013). Speaker Recognition for Mobile User Authentication: An Android Solution. In: *8ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information (SAR SSI)*, pp. 10, France.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, vol. 42, pp. 335–359.

Conneau, A. and Kiela, D. (2018). SentEval: An evaluation toolkit for universal sentence representations. In: Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Cramer, A. L., Wu, H.-H., Salamon, J., and Bello, J. P. (2019). Look, listen, and learn more: Design choices for deep audio embeddings. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856.

de Lope, J. and Graña, M. (2023). An ongoing review of speech emotion recognition. *Neurocomputing*, vol. 528, pp. 1–11.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics.

Ding, S., Chen, T., Gong, X., Zha, W., and Wang, Z. (2020). Autospeech: Neural architecture search for speaker recognition.

Dominguez-Morales, J. P., Liu, Q., James, R., Gutierrez-Galan, D., Jimenez-Fernandez, A., Davidson, S., and Furber, S. (2018). Deep spiking neural network model for time-variant signals classification: a real-time speech recognition approach. In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Ekman, P. (1999). Basic emotions. In: Dalgleish, T. and Powers, M. J., editors, *Handbook of Cognition and Emotion*, pp. 4–5. Wiley.

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., and Truong, K. P. (2016). The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, vol. 7, pp. 190–202.

Eyben, F., Wöllmer, M., and Schuller, B. (2010). Opensmile: The munich versatile and fast open-source audio feature extractor. In: *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 1459–1462, New York, NY, USA. Association for Computing Machinery.

Geetha, A., Mala, T., Priyanka, D., and Uma, E. (2024). Multimodal emotion recognition with deep learning: Advancements, challenges, and future directions. *Information Fusion*, vol. 105.

Ghriss, A., Yang, B., Rozgic, V., Shriberg, E., and Wang, C. (2022). Sentiment-aware automatic speech recognition pre-training for enhanced speech emotion recognition. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2022-May, pp. 7347–7351.

Giannakopoulos, T. (2015). pyaudioanalysis: An open-source python library for audio signal analysis. *PLOS ONE*, vol. 10, pp. 1–17.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Adaptive computation and machine learning. MIT Press.

Grosman, J. (2021). Fine-tuned XLSR-53 large model for speech recognition in English. https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., and Wilson, K. (2017). Cnn architectures for large-scale audio classification. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135. IEEE Press.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, vol. 9, pp. 1735–1780.

Hsu, W., Bolte, B., Tsai, Y. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, vol. abs/2106.07447.

Huang, X., Acero, A., and Hon, H.-W. (2001). *Spoken language processing: a guide to theory, algorithm, and system development*. Prentice Hall PTR, Upper Saddle River, NJ.

Ispas, A.-R., Deschamps-Berger, T., and Devillers, L. (2023). A multi-task, multi-modal approach for predicting categorical and dimensional emotions. In: *ACM International Conference Proceeding Series*, pp. 311 – 317.

Julião, M., Abad, A., and Moniz, H. (2020). Exploring text and audio embeddings for multi-dimension elderly emotion recognition. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2020-October, pp. 2067–2071.

Koh, E. S. and Dubnov, S. (2021). Comparison and analysis of deep audio embeddings for music emotion recognition. *CoRR*, vol. abs/2104.06517.

Koolagudi, S. G., Murthy, Y. V. S., and Bhaskar, S. P. (2018). Choice of a classifier, based on properties of a dataset: case study-speech emotion recognition. *International Journal of Speech Technology*, vol. 21, pp. 167–183.

Labied, M., Belangour, A., Banane, M., and Erraissi, A. (2022). An overview of automatic speech recognition preprocessing techniques. In: *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, pp. 804–809.

Lech, M., Stolar, M., Best, C., and Bolia, R. (2020). Real-time speech emotion recognition using a pre-trained image classification network: Effects of bandwidth reduction and companding. *Frontiers in Computer Science*, vol. 2.

Leow, C. S., Hayakawa, T., Nishizaki, H., and Kitaoka, N. (2020). Development of a low-latency and real-time automatic speech recognition system. In: *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, pp. 925–928.

Loderer, K., Gentsch, K., Duffy, M. C., Zhu, M., Xie, X., Chavarría, J. A., Vogl, E., Soriano, C., Scherer, K. R., and Pekrun, R. (2020). Are concepts of achievement-related emotions universal across cultures? a semantic profiling approach. *Cognition and Emotion*, vol. 34, pp. 1480–1488.

Lotfian, R. and Busso, C. (2019). Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing*, vol. 10, pp. 471–483.

MacAry, M., Tahon, M., Esteve, Y., and Rousseau, A. (2021). On the use of self-supervised pre-trained acoustic and linguistic features for continuous speech emotion recognition. In: *2021 IEEE Spoken Language Technology Workshop, SLT 2021 - Proceedings*, pp. 373–380.

Malik, M., Malik, M. K., Mehmood, K., and Makhdoom, I. (2021). Automatic speech recognition: a survey. *Multimedia Tools and Applications*, vol. 80, pp. 9411–9457.

Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. *Current Psychology*, vol. 14, pp. 261–292.

Munezero, M., Montero, C. S., Sutinen, E., and Pajunen, J. (2014). Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE Transactions on Affective Computing*, vol. 5, pp. 101–111.

Nagrani, A., Chung, J. S., and Zisserman, A. (2017). VoxCeleb: A large-scale speaker identification dataset. In: *Interspeech 2017*. ISCA.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In: Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 689–696. Omnipress.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An asr corpus based on public domain audio books. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210.

Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Pham, N. T., Dang, D. N. M., Pham, B. N. H., and Nguyen, S. D. (2023). Server: Multi-modal speech emotion recognition using transformer-based and vision-based embeddings. In: *Proceedings of the 2023 8th International Conference on Intelligent Information Technology*, ICIIT '23, pp. 234–238, New York, NY, USA. Association for Computing Machinery.

Picard, R. W. (1997). *Affective Computing*. MIT Press, Cambridge, MA.

Pérez-Toro, P. A., Rodríguez-Salas, D., Arias-Vergara, T., Klumpp, P., Schuster, M., Nöth, E., Orozco-Arroyave, J. R., and Maier, A. K. (2022). Interpreting acoustic features for the assessment of alzheimer's disease using forestnet. *Smart Health*, vol. 26, pp. 100347.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., and Bengio, Y. (2020). Multi-task self-supervised learning for robust speech recognition.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Roberts, K., Roach, M. A., Johnson, J., Guthrie, J., and Harabagiu, S. M. (2012). EmpaTweet: Annotating and detecting emotions on Twitter. In: Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 3806–3813, Istanbul, Turkey. European Language Resources Association (ELRA).

Roger, V., Farinas, J., and Pinquier, J. (2022). Deep neural networks for automatic speech processing: a survey from large corpora to limited data. *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, pp. 19.

Russell, J. (1980). A circumplex model of affect. *Journal of personality and social psychology*, vol. 39, pp. 1161–1178.

Saeki, T., Takamichi, S., and Saruwatari, H. (2021). Low-latency incremental text-to-speech synthesis with distilled context prediction network. In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 749–756.

Scherer, K. R. (2005). What are emotions? and how can they be measured? *Social Science Information*, vol. 44, pp. 695–729.

Schuller, B., Steidl, S., Batliner, A., Hirschberg, J., Burgoon, J. K., Baird, A., Elkins, A., Zhang, Y., Coutinho, E., and Evanini, K. (2016). The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language. In: *Proc. Interspeech 2016*, pp. 2001–2005.

Shor, J., Jansen, A., Maor, R., Lang, O., Tuval, O., de Chaumont Quitry, F., Tagliasacchi, M., Shavitt, I., Emanuel, D., and Haviv, Y. (2020). Towards learning a universal non-semantic representation of speech. In: *Interspeech 2020*. ISCA.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Singh, R., Yadav, H., Sharma, M., Gosain, S., and Shah, R. R. (2019). Automatic speech recognition for real-time systems. In: *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pp. 189–198.

Singh, Y. B. and Goel, S. (2022). A systematic literature review of speech emotion recognition approaches. *Neurocomputing*, vol. 492, pp. 245–263.

Smith, S. W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. vol. 2018-April, pp. 5329 – 5333.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In: Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S., editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Sogancioglu, G., Verkholyak, O., Kaya, H., Fedotov, D., Cadée, T., Salah, A., and Karpov, A. (2020). Is everything fine, grandma? acoustic and linguistic modeling for robust elderly speech emotion recognition. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2020-October, pp. 2097–2101.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Srinivasan, S., Huang, Z., and Kirchhoff, K. (2022). Representation learning through cross-modal conditional teacher-student training for speech emotion recognition. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2022-May, pp. 4298–4302. cited By 0.

Stolar, M. N., Lech, M., Bolia, R. S., and Skinner, M. (2017). Real time speech emotion recognition using rgb image classification and transfer learning. In: *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–8.

Sun, L., Lian, Z., Tao, J., Liu, B., and Niu, M. (2020). Multi-modal continuous dimensional emotion recognition using recurrent neural network and self-attention mechanism. In: *Proceedings of the 1st International on Multimodal Sentiment Analysis in Real-Life Media Challenge and Workshop*, MuSe'20, pp. 27–34, New York, NY, USA. Association for Computing Machinery.

Testa, B., Xiao, Y., Sharma, H., Gump, A., and Salekin, A. (2023). Privacy against real-time speech emotion detection via acoustic adversarial evasion of machine learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7.

Triantafyllopoulos, A., Reichel, U., Liu, S., Huber, S., Eyben, F., and Schuller, B. W. (2023). Multistage linguistic conditioning of convolutional layers for speech emotion recognition. *Frontiers in Computer Science*, vol. 5.

Triantafyllopoulos, A., Wagner, J., Wierstorf, H., Schmitt, M., Reichel, U., Eyben, F., Burkhardt, F., and Schuller, B. (2022). Probing speech emotion recognition transformers for linguistic knowledge. In: *Proc. Interspeech 2022*, vol. 2022-September, pp. 146–150.

Van Houdt, G., Mosquera, C., and Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, vol. 53, pp. 5929–5955.

Wagner, J., Triantafyllopoulos, A., Wierstorf, H., Schmitt, M., Burkhardt, F., Eyben, F., and Schuller, B. W. (2023). Dawn of the transformer era in speech emotion recognition: Closing the valence gap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 10745–10759.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020). MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Wang, Y., Song, W., Tao, W., Liotta, A., Yang, D., Li, X., Gao, S., Sun, Y., Ge, W., Zhang, W., and Zhang, W. (2022). A systematic review on affective computing: emotion models, databases, and recent advances. *Information Fusion*, vol. 83-84, pp. 19–52.

Wani, T. M., Gunawan, T. S., Qadri, S. A. A., Kartiwi, M., and Ambikairajah, E. (2021). A comprehensive review of speech emotion recognition systems. *IEEE Access*, vol. 9, pp. 47795–47814.

Williams, C. E. and Stevens, K. N. (1972). Emotions and Speech: Some Acoustical Correlates. *The Journal of the Acoustical Society of America*, vol. 52, pp. 1238–1250.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br