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

LARISSA DAIANE CANEPPELE GUDER

DIMENSIONAL SPEECH EMOTION RECOGNITION: A BIMODAL APPROACH

Porto Alegre
2024

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

DIMENSIONAL SPEECH EMOTION RECOGNITION: A BIMODAL APPROACH

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. PhD. Dalvan Jair Griebler
Co-Advisor: Prof. PhD. João Paulo de Souza Aires

Porto Alegre
2024

Ficha Catalográfica

G922d Guder, Larissa Daiane Caneppele

Dimensional Speech Emotion Recognition : a Bimodal Approach / Larissa Daiane Caneppele Guder. – 2024.
60p.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Dalvan Jair Griebler.
Coorientador: Prof. Dr. João Paulo de Souza Aires.

1. Speech Emotion Recognition. 2. Natural Language Processing. 3. Streaming. I. Griebler, Dalvan Jair. II. Aires, João Paulo de Souza. III. , . IV. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

LARISSA DAIANE CANEPPELE GUDER

DIMENSIONAL SPEECH EMOTION RECOGNITION: A BIMODAL APPROACH

This Master Thesis has been submitted in partial fulfillment of the requirements for the degree of Master in Computer Science of the Computer Science Graduate Program, School of Technology of the Pontifical Catholic University of Rio Grande do Sul

Sanctioned on March 22, 2024.

COMMITTEE MEMBERS:

Prof. PhD. Renata Vieira (Universidade de Évora)

Prof. PhD. Duncan Ruiz (PPGCC/PUCRS)

Prof. PhD. João Paulo de Souza Aires (PPGCC/PUCRS- Co-Advisor)

Prof. PhD. Dalvan Jair Griebler (PPGCC/PUCRS - Advisor)

Dedicated to my emotional support: Cristina and Maria.

“No fim, é disso de que somos feitos: sons, palavras, emoções e um punhado de memórias.”
(Mírian Maranhão)

ACKNOWLEDGMENTS

First and foremost, I am deeply grateful to Dr. Dalvan for not only opening the doors of the PUCRS to me but also for his unwavering support and guidance throughout my master's program. Even without being on campus in person, I had all the support I needed to stay here.

To João, for guiding me in the process. Thank you so much for taking the time to review my texts and for being there to lend an empathetic ear to all my complaints.

To GMAP members, especially to Higor, Luan, and Arthur.

To SAP for the financial support.

To my family, thank you for all the support during my master's.

In particular, I would like to thank my psychology tripod. Although I worked with emotion recognition, I could not deal with my emotions without Kelly's help. Finally, I would like to express my gratitude to Cristina and Maria. They have been with me since the time I decided to sign up, even before them. They made the entire process much easier for me and did not allow me to cancel my enrollment. I appreciate your help and I feel truly blessed to have you in my life.

RECONHECIMENTO DIMENSIONAL DE EMOÇÕES NA FALA: UMA ABORDAGEM BIMODAL

RESUMO

Considerando a relação humano-computador, a computação afetiva visa permitir com que computadores sejam capazes de reconhecer ou expressar emoções. O Reconhecimento de Emoções na Fala é uma tarefa da computação afetiva que tem como objetivo reconhecer emoções presentes em um segmento de áudio. O modo tradicional de predizer emoções na fala é utilizando classes pré-determinadas, no modo offline. Dessa maneira, o número de emoções que pode ser reconhecido é limitado ao número de classes. Para evitar essa limitação o reconhecimento dimensional de emoção utilizando dimensões como a valência, ativação e dominancia, consegue representar emoções com maior granulidade. Pesquisas recentes propõem o uso de informações textuais para melhorar os resultados da valência. Apesar dos esforços recentes para tentar melhorar os resultados no reconhecimento dimensional de emoções na fala, eles não consideram cenários do mundo real, onde é necessário processar a entrada em um curto espaço de tempo. Considerando estes aspectos, nesse trabalho, são dados os primeiros passos através de uma abordagem bimodal para o reconhecimento dimensional de emoções na fala em streaming. Nossa abordagem combina representações de sentenças e áudio como entrada para uma rede neural recorrente, que realiza o reconhecimento de emoções na fala. Nós avaliamos diferentes métodos para criar as representações de texto e de áudio, bem como técnicas para o reconhecimento automático da fala. Nossos melhores resultados atingiram 0.5915 de CCC de ativação, 0.4165 para valência, e 0.5899 de dominância no dataset IEMOCAP.

Palavras-Chave: Reconhecimento de Emoções na Fala, Processamento de Linguagem Natural, Streaming.

DIMENSIONAL SPEECH EMOTION RECOGNITION: A BIMODAL APPROACH

ABSTRACT

Considering the human-machine relationship, affective computing aims to allow computers to recognize or express emotions. Speech Emotion Recognition is a task from affective computing that aims to recognize emotions in an audio utterance. The most common way to predict emotions from the speech is using pre-determined classes in the offline mode. On that way, the emotion recognition is restricted to the number of classes. To avoid this restriction, dimensional emotion recognition uses dimensions such as valence, arousal, and dominance, can represent emotions with higher granularity. Existing approaches propose using textual information to improve results for the valence dimension. Although recent efforts try to improve results on Speech Emotion Recognition to predict emotion dimensions, they do not consider real-world scenarios, where processing the input in a short time is necessary. Considering these aspects, in this work, we give the first step towards creating a bimodal approach for Dimensional Speech Emotion Recognition in streaming. Our approach combines sentence and audio representations as input to a recurrent neural network that performs speech emotion recognition. We evaluate different methods for creating audio and text representations, as well as automatic speech recognition techniques. Our best results achieve 0.5915 of CCC for arousal, 0.4165 for valence, and 0.5899 for dominance in the IEMOCAP dataset.

Keywords: Speech Emotion Recognition, Natural Language Processing, Streaming.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AED – Audio Set Acoustic Event Detection
ASR – Automatic Speech Recognition
BERT – Bidirectional Encoder Representations from Transformers
CCC – Concordance Correlation Coefficient
CER – Character Error Rate
CNN – Convolutional Neural Networks
DCT – Discrete Cosine Transform
DL – Deep Learning
DNN – Deep Neural Networks
F0 – Fundamental Frequency
FFT – Fast-Fourier-Transformation
GEMAPS – Geneva Minimalistic Acoustic Parameter Set
GLOVE – Global Vectors for Word Representation
HUBERT – Hidden unit BERT
IEMOCAP – The Interactive Emotional Dyadic Motion Capture
LLDS – Low-level descriptors
LSTM – Long Short-Term Memory
MFCCS – Mel-Frequency Cepstral Coefficients
ML – Machine Learning
MSE – Mean Squared Error
NLP – Natural Language Processing
PAA – pyAudioAnalisys
PCA – Principal Component Analysis
RNN – Recurrent Neural Network
SBCAS – Brazilian Symposium on Computing Applied to Health
SBERT – Sentence BERT
SER – Speech Emotion Recognition
SVM – Support Vector Machines
TRILL – TRIpLet Loss network
UWA – Unweighted Accuracy
WER – Word Error Rate

LIST OF SYMBOLS

CONTENTS

1. INTRODUCTION

Our emotions play a subjective and controversial role, vital to our psychic survival. Understanding, to a certain extent, the emotions of other people and how they express them is fundamental to relating to each other as a society. For example, while fear is a natural protective regulator and aids decision-making, anger allows us to set limits and develop our sense of justice. An example of the importance of understanding emotions is that in autistic people, persistent deficits in emotional reciprocity and non-verbal communication, along with other factors, can lead to greater difficulty in communication and social interaction [5]. Based on this, emotion recognition is more a perspective than an exact science.

Besides the ways used to determine emotions in psychology, two approaches have been used to recognize emotions using deep learning: discrete classes and dimensional [43]. In discrete classes, the six emotions considered essential by Ekman [22]: anger, disgust, fear, happiness, sadness, and neutral are used, where the model must classify the input according to the most correlated class. On the other hand, Russell [61] defines a dimensional approach through the circumplex model of affect. The circumplex model considers two dimensions: arousal and valence. Each dimension has a value that ranges from -1 to 1. Arousal is related to calming or exciting the tonality of speech, while valence represents how pleasant or not it is. With the score of each dimension, it is possible to correlate to a specific emotion. For example, fear and anger can be defined with low valence and high arousal. Mehrabian [47] adds the dominance dimension, representing how emotion influences a person's behavior. It is important that models recognize emotions and respect each person's idiosyncratic diversity.

Leaving the direct application in psychology, different sectors benefit from recognizing emotions daily. The review by Geetha *et al.* [26] 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.* [77].

The lack of data for training and testing deep learning models makes it difficult for the field of SER to grow [18]. Existing datasets have a small amount of available data, are less diverse than necessary, or are too different from real-world data. Even when focusing only on speech emotion recognition, it is necessary to consider that human emotion perception involves multiple senses, being multimodal [26]. So, to overcome, and extract more information from only spoken data, the use of textual information can improve the precision of the predictors. Some authors have already shown that using text features, such as word embeddings, improves valence prediction [79, 74, 27, 6, 7, 72, 37]. However, including new features in the processing usually increases the time necessary

to generate an output. For instance, including text features requires first transcribing the audio to use it as input.

This work addresses these challenges through an architecture for speech emotion recognition usable in a streaming environment. Our key contributions are the development of a series of software architectures that overcome these gaps via 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. We empirically show the effectiveness by evaluating 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 for automatic speech recognition. 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. Then, we use a concatenation layer to join the features. In contrast to Atmaja and Akagi [6] 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 it possible for a computer to recognize emotions, we have some research areas focusing on this. Created by Picard [55], 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 human behavior. We have different levels of approaches that focus on understanding human behavior through the recognition of emotions and analysis of sentiments [84].

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.* [49] 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 affect expression. Social and cultural factors influence emotions and represent the preconscious expression of feelings and affect. Otherwise, sentiments are conscious and built over time, considering social influence. Finally, opinion is more related to how each person interprets information, considering emotions or not.

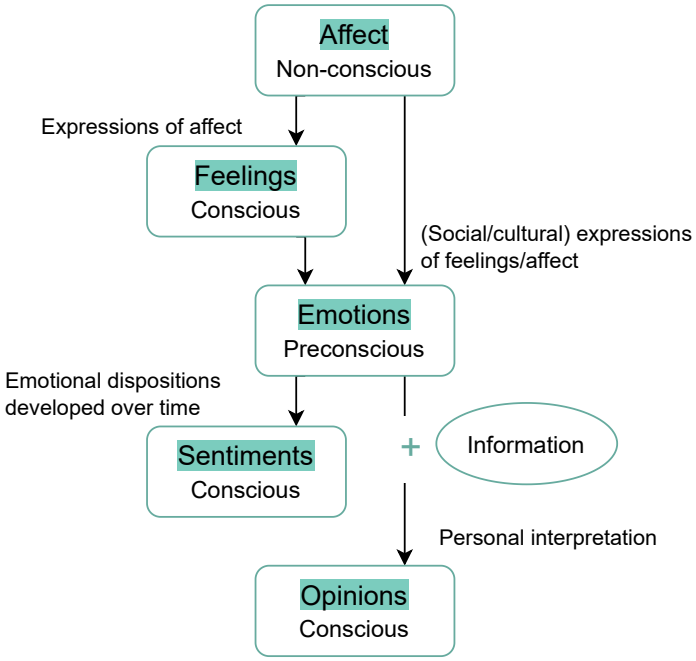

Figure 2.1 – Differentiating factors between affect, feelings, emotions, sentiments, and opinions. Adapted from Munezero *et al.* [49]

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.* [13] defines that emotions are culturally grounded and dynamically experienced, to some extent, constructed in interaction. Complementary to that, Loderer *et al.* [44] states that emotions can be perceived differently in different cultures. Loderer *et al.* [44] findings show that the more similar components across cultures are affective, cognitive, and motivational. On the other hand, the less similar are physiological and expressive components.

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 [55].

Figure 2.1 details the structure of an emotion. At the top, we have the culture-independent level, representing how the individual perceives emotions. Then, we have the culture and society-dependent level, which describes how individuals name their feelings.

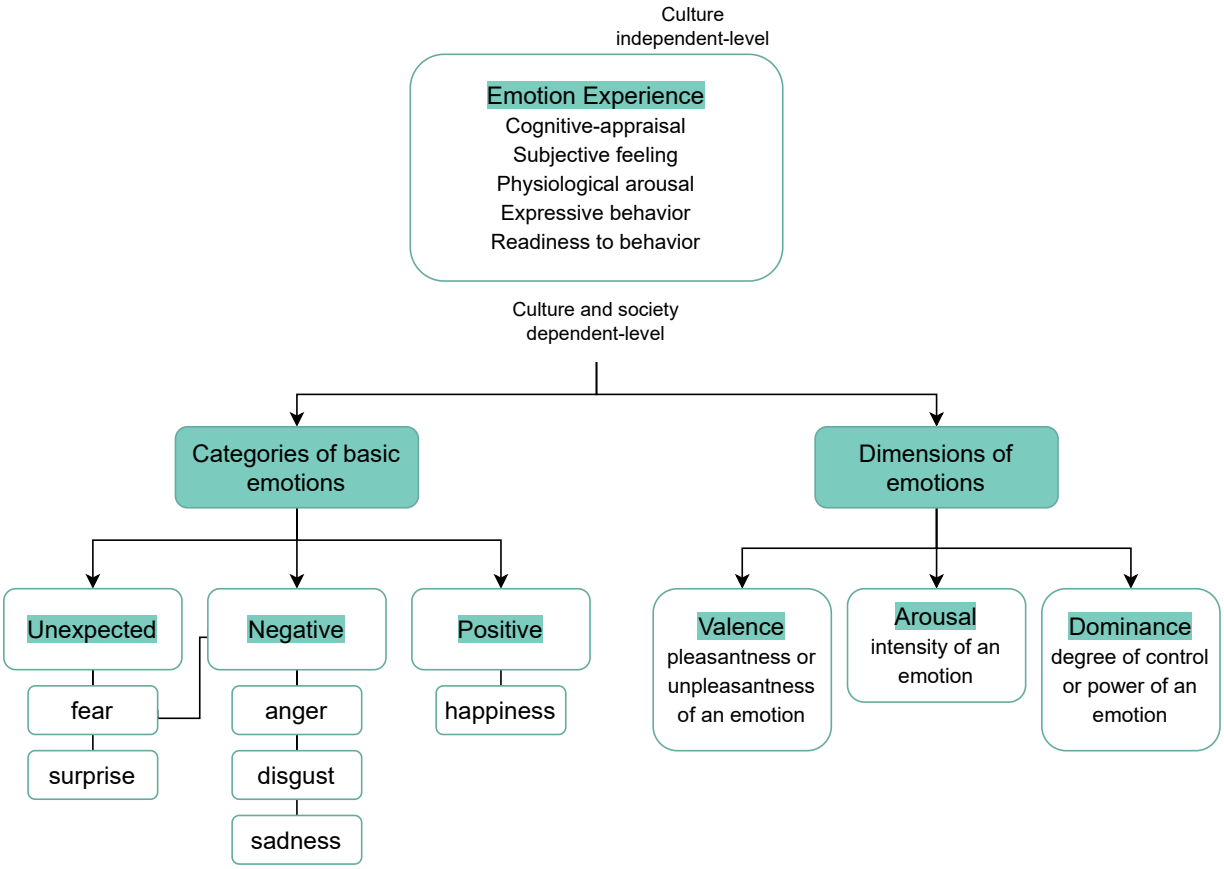

Figure 2.2 – Structure of emotion experience and classification, adapted from Munezero *et al.* [49] and Roberts *et al.* [59]

To make it possible for a computer to recognize an emotion, it is necessary to classify it mathematically. Applied to machine learning, two classifications are often used:

discrete classification and dimensional classification [43]. Ekman [22] 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 [61], which we will call dimensional classification. Two axes represent values for arousal (y-axis) and valence (or pleasure) (x-axis) in a dimensional space. These two values range from -1 to 1, making it 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 [63], which has more emotions mapped than the original version proposed by Russell [61]. 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 [47] 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 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

Speech is one of the bases of human communication. Through speech, we can transmit information and express our emotions. Audio processing is a subfield of Digital Signal Processing that converts sound into a format that machines can process. 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 [69].

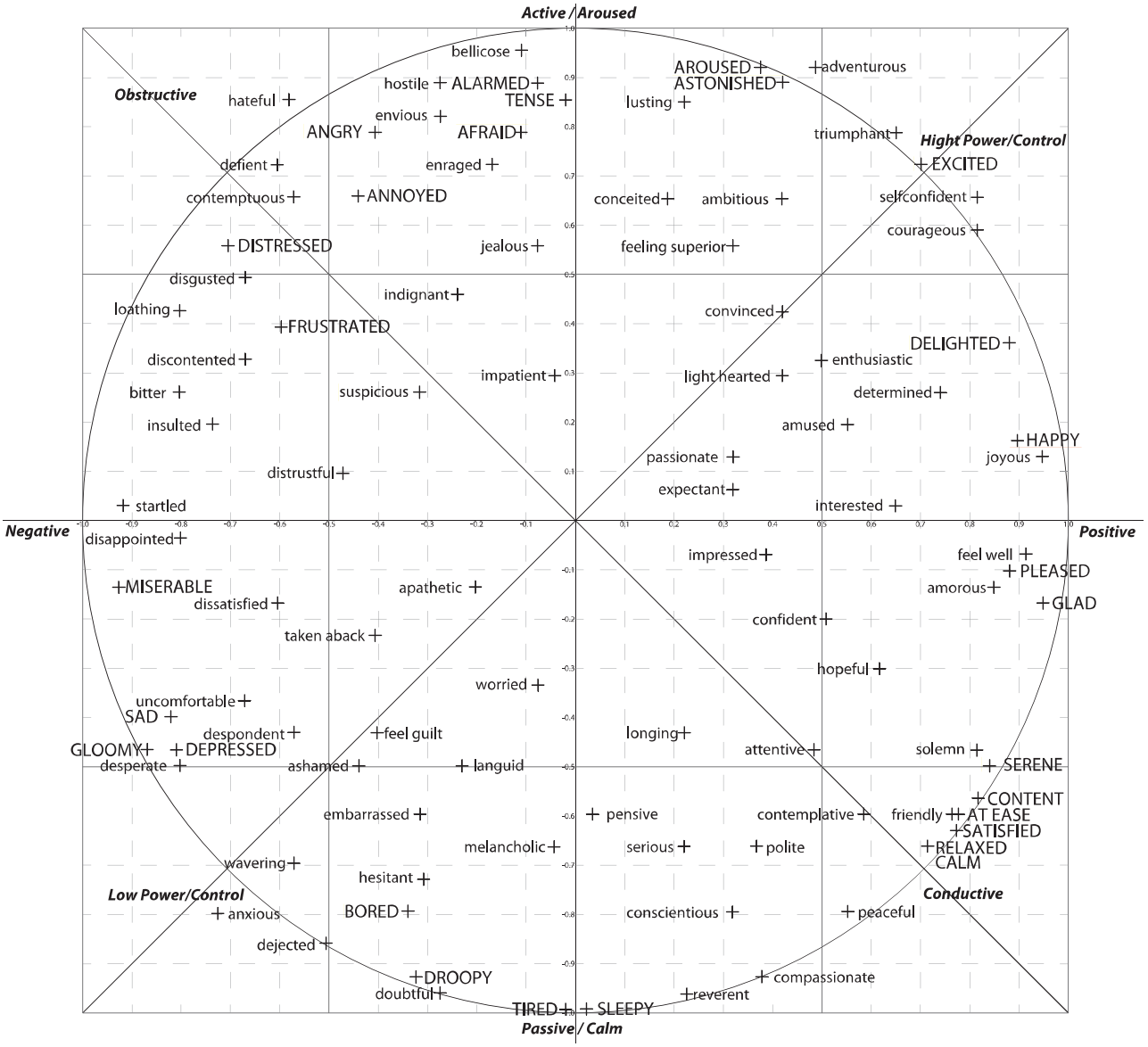

Figure 2.3 – Updated version from [61] circumplex model of affect, proposed by Scherer [63], focusing on the semantic space for emotions. Adapted from Ahn *et al.*[1].

One of the tasks in audio processing is emotion recognition. First, we need to extract features from the low-level descriptors (LLDs) to make it possible to recognize emotions in speech. LLDs provide ways to extract information from the digital signal. They can be grouped into three domains: prosodic, spectral, and voice quality.

Considering the prosodic domain, there are three most commonly used: fundamental frequency or pitch, energy, and duration. The Fundamental Frequency, or F0 or pitch, opens and closes the vocal folds in phonation [35]. Williams and Stevens [85] 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 [6], MacAry *et al.* [45], and Julião *et al.* [37] still use it. Energy represents how loud or intense a sound signal can be, it is measured in decibels (dB). Energy is directly related to the arousal dimension, indi-

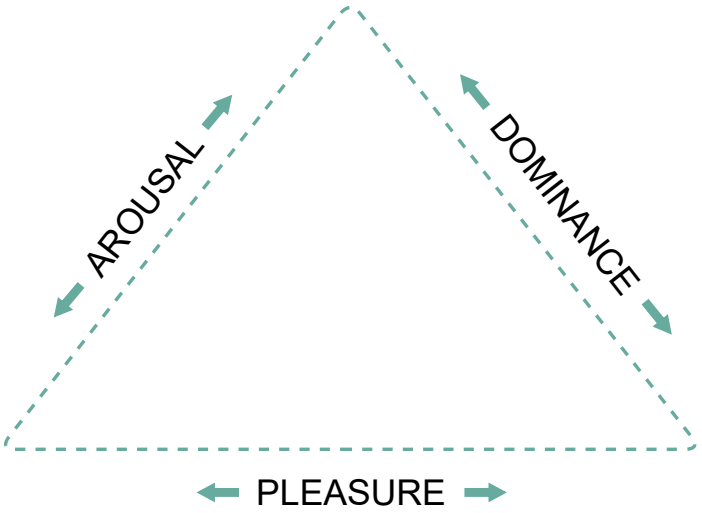

Figure 2.4 – Pleasure-Arousal-Dominance Emotional State Model proposed by Mehrabian [47]

cating how calm or energic the speech is. 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). MFCCs, in a general way, are used to translate for a machine how humans perceive sounds. Huang *et al.*[35] 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.

Figure 2.5 illustrates MFCC's steps 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 [14]. 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.

Related to voice quality features, we have shimmer and jitter. 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 [39]. Jitter is calculated through Equation 2.1, where T_i is the considered pitch period, and N is the number of cycles.

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

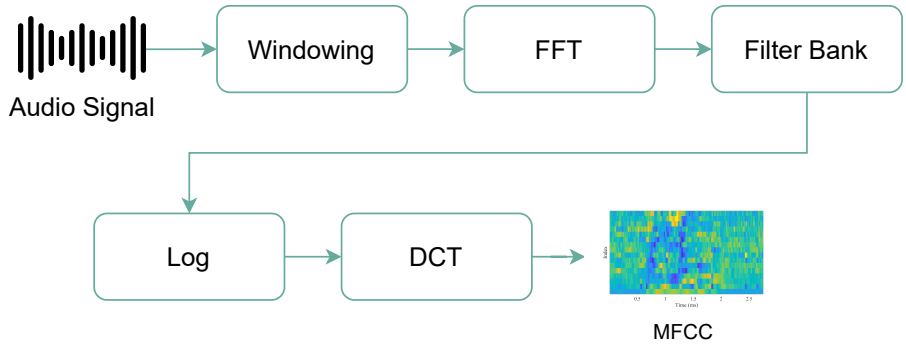

Figure 2.5 – MFCC process

On the other hand, shimmer can calculate the energy variation. Koolagudi *et al.*[39] 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. Focusing on determining which combinations of features are best to be used in some tasks that involve speech processing, such as SER, we have some feature sets like eGeMAPs [23] and ComParE [64]. 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 LLDs and some statistical functionals, like the arithmetic mean and coefficient of variation over the LLDs.

To extract these feature sets from audio, there are two main Python libraries: OpenSmile [24] and pyAudioAnalisys (pAA) [28]. 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 2.1. 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 [24].

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 [24]

pAA is an open-source option for extracting features. Table 2.2 shows the complete features and descriptions from short-term extraction. The features detailed in the table can be extracted 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 [65] and VGGish [32]. 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 uses a variant of the ResNet-50 with a 512-dimensional embedding layer [65].

VGGish [32] is a modification of the VGG16 architecture [66], 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.

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 [28]

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 [29]. Malik *et al.* [46] 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 preprocessing are voice activity detection, noise removal, pre-emphasis, framing, windowing, and normalization [40].

After cleaning the audio input, extracting the input features for the model is necessary. 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

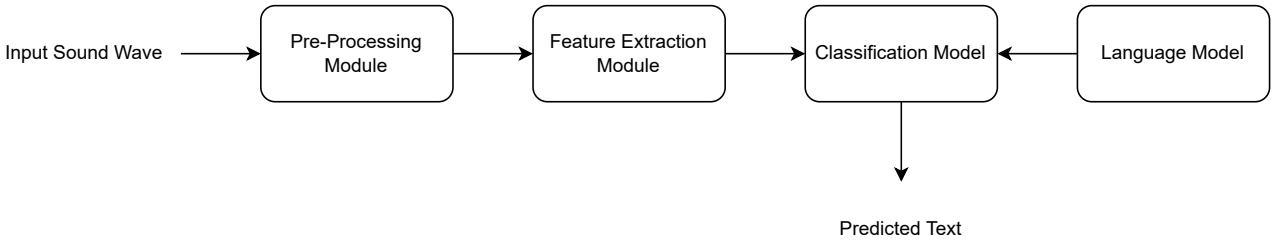

Figure 2.6 – Basic structure of an ASR, adapted from Malik *et al.* [46]

domain. For ASR, it is common to use MFCC, PLP, DWT, relative spectral-perceptual linear prediction (RASTA-PLP), and LPC [46].

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) [46].

The typical metrics used to evaluate the classifier 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.

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

Surveys such as the one conducted by [60] list the most used datasets in the ASR task. The datasets are: LibriSpeech [52], IEMOCAP [15], and VoxCeleb1 [50]. The state-of-the-art models in these datasets are: Pase+ [57], Wav2Vec2.0 [9], HuBERT [34] and AutoSpeech [20]. Deep learning approaches were recently used, achieving better results [3].

Wav2Vec2 [9] 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 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. 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 six layers.

Hidden unit BERT (HuBERT) [34] 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 [19] 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 [19]. 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 [56].

2.3 Sentence Representation

Representing text as vectors is common in multiple Natural Language Processing (NLP) tasks. These vectors often describe text features, such as the presence of specific words, their frequency in a text, or their semantic meaning. We use vectors to represent different granularity of a text, such as words, sub-words, sentences, and paragraphs. Recent approaches introduce the use of learning models to generate these vectors. These models can capture both the syntactic and semantic information of texts by considering their context. Sentence representations or embeddings are a result of this process, they represent sentences numerically through vectors in a high-dimensional space. This representation keeps the semantic relationship and makes it possible to extract the meaning of a sentence. Since emotion recognition depends on the context to make sense, sentence embeddings can be a great alternative to this.

We can use specific models, such as Sentence-BERT (SBERT) [58] to generate sentence embeddings. SBERT is a modification of BERT [19], one of the state-of-the-art models for word embedding. Devlin *et al.* [19] 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 [16] 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.

The MPNet [73] model combines the use of Masked Language Modeling (MLM) present on BERT and Permuted Language Model (PLM) in the XLNet model. Random tokens were hidden in the input text with masking, while the permutation randomly reordered these tokens. With this, MPNet learns more robust representations. The model will generate a representation with 768 for sentences. They are trained over 1.170.060.424 training pairs.

Large pre-trained models can have some computational costs to execute. The MiniLM [83] is a task-agnostic and distilled approach focusing on a lightweight version of Transformer-based models. Using the teacher-student architecture, the authors propose a distilled version of the self-attention heads of the teacher to make this possible. We explore two different pre-trained versions: the paraphrase-MiniLM-L3 and all-MiniLM-L12. The all-MiniLM-L12 [1] version is trained over the same train set of the MPNet. On the other hand, the paraphrase MiniLM L3 [2] is a three-layer version of the MiniLM L12. Both produce a representation with 384 dimensions for sentences.

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 [29]. Deep Learning has been used to solve real-world problems in different domains, such as agriculture, psychology, health, and traffic. It 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 [51].

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2
[2] https://huggingface.co/nreimers/MiniLM-L6-H384-uncased

We have two different types of learning methods: supervised and unsupervised learning. In supervised learning, the model learns from labeled data. The model adapts to map input features into the correct labels throughout training. 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 [12].

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 [26].

Long Short-Term Memory (LSTM) was proposed by Hochreiter and Schmidhuber [33] 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 2.7. The architecture is composed of an input gate, which defines the information that will be added to the cell state; a forget gate, which 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, which saves the information that passes through the LSTM. LSTM is a good choice for audio signal processing because it can use information from previous states to compute the result for new ones.

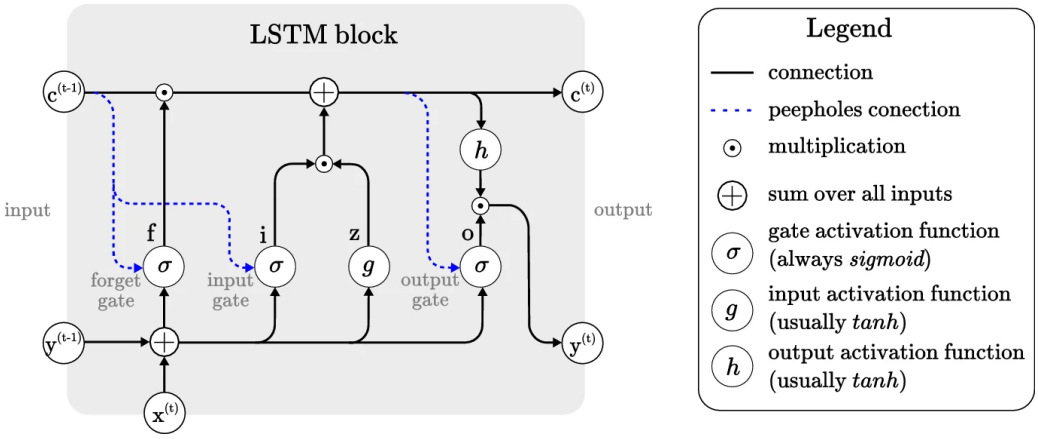

Figure 2.7 – Basic structure of LSTM [80]

2.5 Speech Emotion Recognition

Sing and Goel [68] defines the Speech Emotion Recognition (SER) task as recognizing emotions from speech utterances without using linguistic features. Since speech is one of the most used ways to communicate in society, the SER can be applied in different sectors, like education, healthcare, marketing and advertising, human-robot interaction,

security and surveillance, customer service, sports, entertainment, gaming, and the automotive industry [26]. Also, the use of speech is less intrusive than physiological signals.

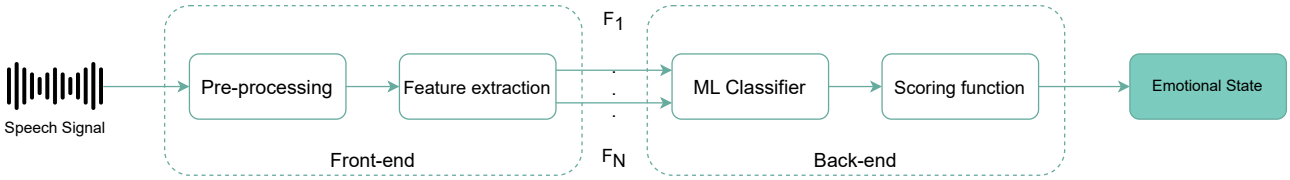

Figure 2.8 – Basic structure of SER, adapted from Lieskovská *et al.* [43]

Lieskovská *et al.* [43] summarize SER into two main parts: feature extraction and classification. In Figure 2.8, we illustrate these two parts. As we can see, the speech signal serves as input to the front-end. In the front-end, we have the preprocessing and feature extraction steps that define a representation for the input signal. This representation can be a pre-defined set of features like eGeMAPs, ComParE, and pAA or audio embeddings, like VGGIsh-generated ones. After generating the representation, they are 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.

In the back-end, two different methods have been applied. One is using CNN models, treating SER as an image classification task, where the image of the Mel Spectogram feeds the model. The other uses models like SVM, decision trees, and autoencoder when features such as prosodic, voice quality, and MFCC are used.

Datasets used to train and evaluate models have three types of origin: actor-based, induced, and natural emotion. Actor-based datasets are developed under laboratory 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 [68].

To evaluate the results predicted with models that use a dimensional approach, the Concordance Correlation Coefficient (CCC) and the Mean Squared Error (MSE) (Equation 2.5) are used. CCC (Equation 2.4) 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}.

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

$$\sum_{i=1}^{D}(x_i - y_i)^2 \tag{2.5}$$

2.6 Data Streaming

Data streaming is a continuous flow of data that, by default, never ends [4]. Nowadays, many streaming applications are running: wearable sensors, traffic information, social media, streaming services such as Spotify and Netflix, autonomous cars, and many others. 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 [4].

The data tuple can be structured, semi-structured, or unstructured. It is 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. Finally, unstructured data tuples consist of data that do not have patterns or are in a proprietary format [4].

The process that will receive and process these tuples is called by Andrade *et al.* [4] as data flow graphs. The operations 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 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, producing 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 is windowed-based, which groups and sorts tuples based on a key value; join that are windowed-based and associates tuples based on a condition; barrier differs from join because they do not use match conditions. The barrier is also used to synchronize streams.

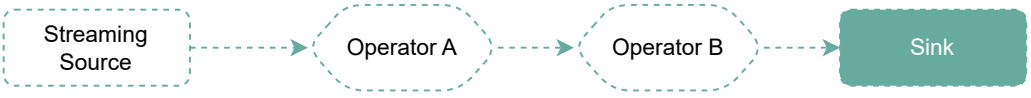

Figure 2.9 – Pipeline example

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.

Related to the data source, we can divide data streaming into two categories: event-based and continuous data. Event-based events are triggered events that occur under certain conditions, such as 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, which can then be fed to 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.

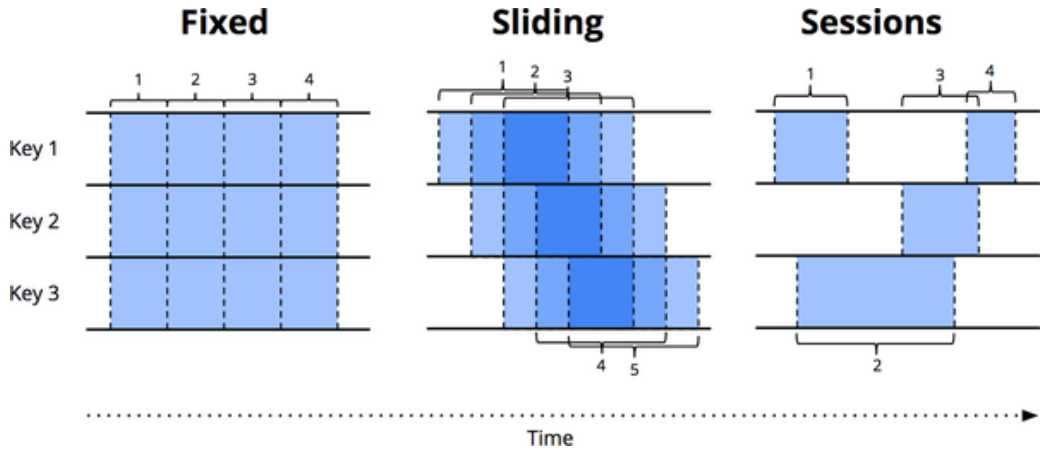

Figure 2.10 – Windowing strategies [2]

Akidau *et al.* [2] 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.

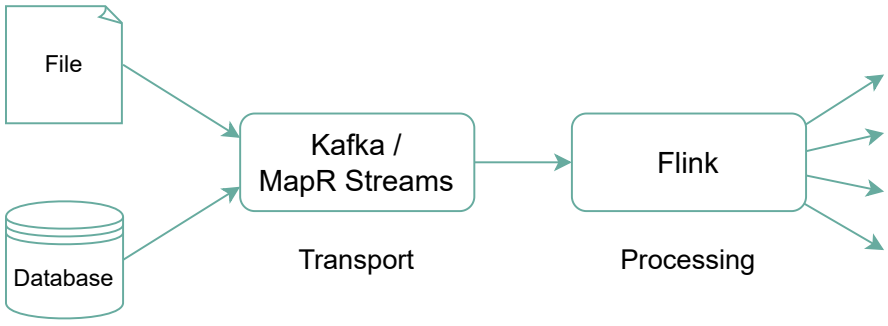

Figure 2.11 – Architecture of flink processing, adapted from Friedman and Tzoumas [25, p. 21]

Flink is a Java-based data-streaming framework that considers the event-driven paradigm. The architecture comprises two main concepts: the transport and the stream

processing system. Transportation is responsible for receiving and sending events from the different input sources to subscribe to the consumer and managing the queue of incoming events. Kakfa and MapR Streams are examples of libraries used for this case. The stream processing system stage manages the data transition between applications, the processing and transformations in data, and keeping the application state [25, p. 21].

3. RELATED WORK

In this section, we introduce some related work found 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 3.1.

We divide the analysis of the related works in two ways: (1) approaches that use dimensional emotion recognition and text features; and (2) approaches that apply 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. For the specific scenario, we have only a few models that use classes and audio-only data.

3.1 Dimensional Speech Emotion Recognition

Sun *et al.* [76] 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-obtained 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.* [76] used IS13 with BERT and achieved 0.4931 of CCC for arousal. In contrast, the authors used PyAudio with BERT-4 for the valence dimension and achieved 0.4633 of CCC.

Atmaja and Akagi [6] 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 two 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 [6] also tested with the CNN network. However, the findings show that the LSTM had better multimodal dimension emotion recognition results.

Sogancioglu *et al.* [72] 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 [53] 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 architecture combined Support Vector Machines (SVM), kernel Extreme Learning Machines (ELM), and Partial Least Squares (PLS). 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.* [37] 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 [7] 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.* [79] focuses on evaluating the impact of using a fine-tuned version of the w2v2-L-emo-ft model from Wagner *et al.* [81] 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.* [74] 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.* [36] 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 the DeBERTaV3 was used for textual. HuBERT and DeBERTaV3 have the same 1024 hidden dimension size; To maintain consistent dimensions, the shorter sequence is padded to match the longer one. Cross-attention involves 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.

Using 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 [79, 74, 6, 7, 72, 37]. We notice the use of GloVe by [6, 7] and more recent approaches, such as BERT [74, 37, 76] and a derivation of it called camemBERT [45], and DeBERTaV3 [36].

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 [38] evaluate L3-Net [17] and VGGish models. For SER, Wang *et al.* [82] explored VGGish, but for categorical evaluation, while for dimensional Julião *et al.* [37] explored the use of x-vectors [70] embedding, Sun *et al.* [76] even evaluated the use of VGGIsh, but not for the bimodal approach. More recent approaches consider the use of w2v2 [79] and HuBERT [36, 74] to generate the representations. Independent of the method to extract the features from the audio, even using pre-trained models or handcrafted options, none had the time to process this information. 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.* [11] 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.* [75] 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.* [41] 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 bandwidth reduction from 8 to 4 kHz decreased the accuracy by 3.3%

Unlike these approaches, we will use dimensional emotion recognition instead of discrete classes. Also, our focus is on bimodal features, while [75], [11] and [41] 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 [21, 67, 42] and [62]. It is important to notice that only [41] provides metrics for evaluating streaming scenarios. [75] and [11] only mentioned that their approaches are in real-time but do not show the result.

Ref	Architecture	Audio Embedding	Sentence Embedding	Dataset Type	Language	Dataset	Dimensions Evaluated	Total Classes	Streaming	Data Type	Year
[11]	CNN	No	No	Natural	English	TED-LIUM	-	6	Yes	Audio	2016
[75]	FTAlexNet	No	No	Acted	German	EMO-DB	-	7	Yes	Audio	2017
[41]	CNN	No	No	Acted	German	EMO-DB	-	7	Yes	Audio	2020
[76]	Self-Attention + LSTM	No	No	Natural	English	MuSe-CaR	AVD	-	No	Audio Text	2020
[6]	LSTM	No	No	Acted	English	IEMOCAP	AVD	-	No	Audio Text	2020
[72]	SVM	No	No	Natural	German	USOMS-e	AV	-	No	Audio Text	2020
[37]	SVM	Yes	No	Natural	German	USOMS-e	AV	-	No	Audio Text	2020
[7]	SVM	No	No	Acted Natural	English	IEMOCAP MSP-PODCAST	AVD	-	No	Audio Text	2021
[79]	w2v2 fine-tuning	Yes	No	Natural	English	MSP-PODCAST	AVD	-	No	Audio Text	2022
[74]	Conditional Teacher-Student	No	No	Natural Acted	English	MSP-PODCAST IEMOCAP	AVD	-	No	Audio Text	2022
[78]	MFCNN14	Yes	No	Natural Acted	English	MSP-PODCAST IEMOCAP	AVD	-	No	Audio Text	2023
[36]	Transformer	Yes	No	Acted	English	IEMOCAP	AVD	-	No	Audio Text	2023
Our Approach	**LSTM**	**Yes**	**Yes**	**Acted**	**English**	**IEMOCAP**	**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, and determine the level of customer satisfaction, among other things. However, there is a gap between the literature and the real world, in which we have many approaches for SER, but no one is built to support real-world scenarios with processing information as soon as they are available. 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 combine SER, deep learning, and streaming to build a robust approach that can be applied to the real world.

4.1 End-to-End Speech Emotion Recognition Architecture

Our end-to-end architecture is composed of two blocks. The front-end and the back-end. The front-end is responsible for extracting features from the input signal, while the back-end is responsible for processing the information from the front-end and predicting the output. We detail the architecture in Figure 4.1. Given raw audio, we transform it into a mono waveform and resample it into a 16 kHz sample rate. Due to the VGGIsh input limitation, we limit the audio length to 10 seconds. We extract two types of features from the waveform: textual and acoustic.

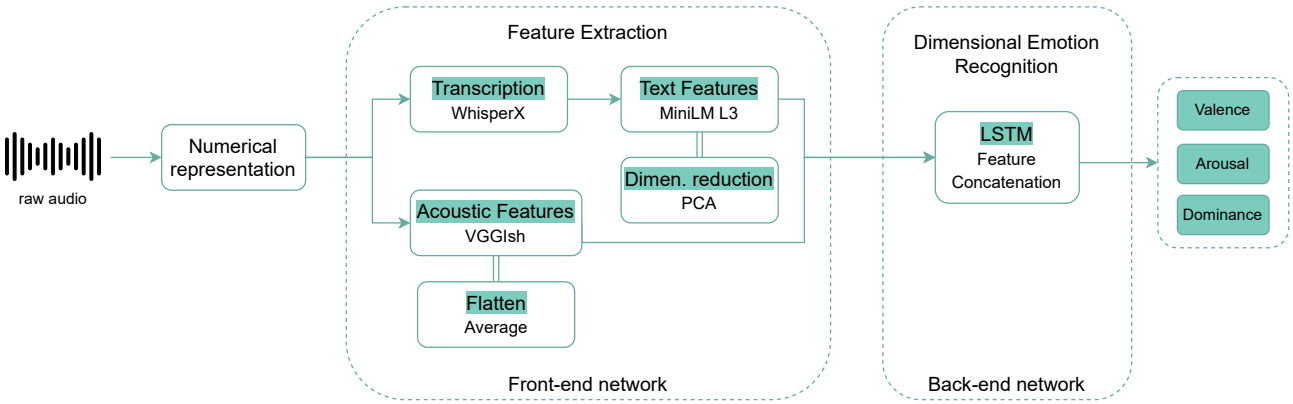

Figure 4.1 – End-to-End Speech Emotion Recognition Architecture

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 so that it can 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 calculate the

average from all rows in the matrix as a flattened function, generating a unique vector with 128 dimensions for the whole audio 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 sentence embedding to be generated for textual representation. To generate the sentence embedding, we use the MiniLM L3 pre-trained model that generates a vector with 384 dimensions. To match the same size as the audio features, we use the Principal Component Analysis (PCA) algorithm to reduce the dimension to 128.

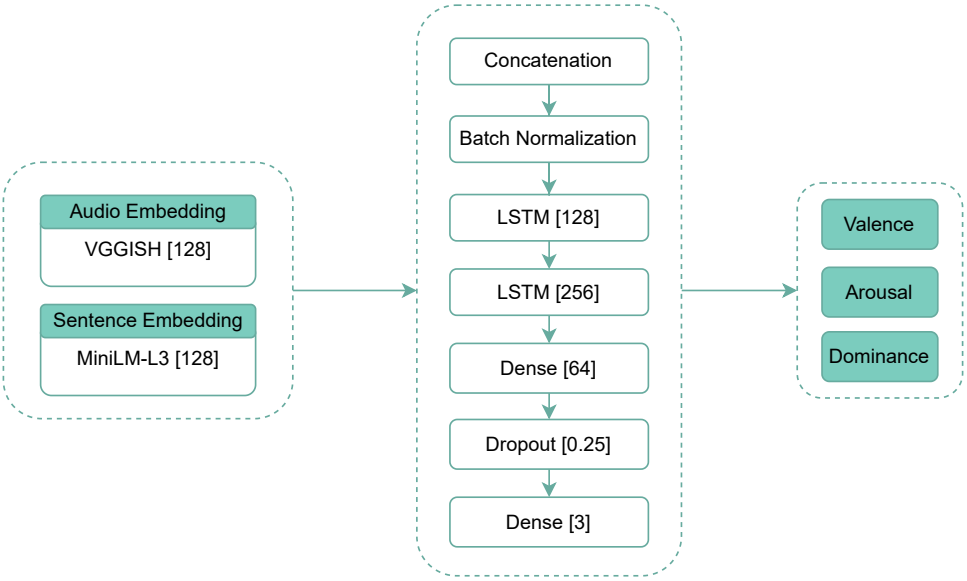

Figure 4.2 – Back-end Architecture

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 after the dense layer. The output is a dense layer with three values corresponding to valence, arousal, and dominance dimensions. We use *tanh* as the activation function and Adam optimizer with a 0.001 learning rate.

4.2 Evaluation Results

We train and evaluate our model on the IEMOCAP dataset. We further detail on Section 5.1. On IEMOCAP, we used the solution provided by Atmaja and Akagi [6] as a baseline to compare our approach. As detailed in Chapter 3, Atmaja and Akagi [6] 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 [6] 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) [71]. The textual embedding focuses on improving the valence dimension; the task is close to sentiment analysis, going from negative to positive perspectives.

Mode	CCC/MSE			AVG
	Valence	Arousal	Dominance	
Baseline				
Bimodal LSTM (GloVe + HSF from pAA) [6]	**0.418**	0.571	0.500	0.496
Our approach				
VAD MiniLM-L3 VAD	0.4165	0.2989	0.2989	0.3381
LSTM Concat (VGGISH + MiniLM-L3 PCA) VAD	0.1431	**0.5915**	**0.5899**	0.4415

Table 4.1 – IEMOCAP evaluation results

On the acoustic side, the use of VGGIsh to recognize emotions has been explored by Pham *et al.* [54] in bimodal categorical speech emotion recognition and by Koh and Dubnov [38] in music emotion recognition. Pham *et al.* [54] 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 2.97ms per second of audio, while the approach of Atmaja and Akagi [6] uses pAA with 9.13ms per second (see Table 5.3). Analyzing the best scenario for each dimension, on valence, we have a loss of 0,359% of CCC in relation to baseline, while for arousal, we have a gain of 3.59%, and for dominance, 17.98%.

5. EXPERIMENTS

Throughout this research, we performed experiments to identify the best architecture for the proposed task and the best feature sets to use with the model. We detail the process to define the architecture in Figure 5.1. To conduct the experiments, we divide the process into two main steps: (1) feature selection and (2) fusion approaches. 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 Datasets

To evaluate our experiments, we use the IEMOCAP (The Interactive Emotional Dyadic Motion Capture) dataset [15]. 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 an AVD 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.

We normalized to a -1 to 1 scale with the Equation 5.1. This normalization is since the original Russel approach uses the -1 to 1 scale, which is the pattern we use in our final architecture.

$$\frac{x - \left(\frac{max - min}{2} + 1\right)}{\frac{max - min}{2}} \tag{5.1}$$

5.2 Feature Selection

To perform SER in a streaming scenario, optimal libraries or models must be chosen to generate the representations of the input sources. 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 a representation for the acoustic information, and (3) generating sentence embeddings for textual information. This set of experiments is detailed in Figure 5.1-[A]. We use the IEMOCAP dataset to compare different approaches. Experiments 1 and 3 use the full dataset, while experiment 2

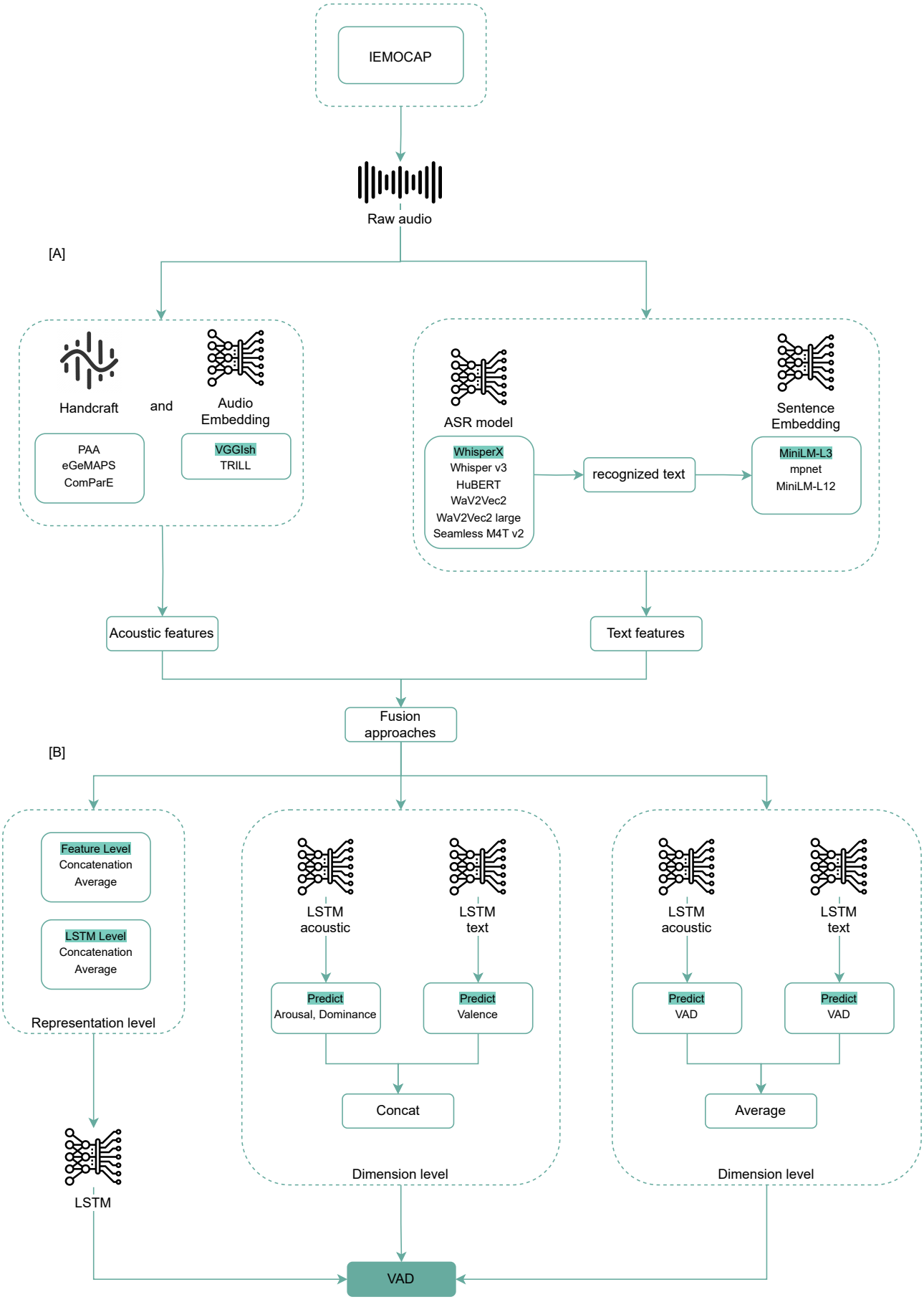

Figure 5.1 – The complete process for speech emotion recognition framework

uses only the test set. The objective for each one is to define the best option, considering the speed at which the data is processed and the lower error rate on evaluation.

We consider only pre-trained models for (1) ASR task. To make the transcription, we select state-of-the-art models and compare the speed (time used to transcribe a chunk of audio) and the Word Error Rate (WER) to measure the quality of the transcribed text. In our tests, we evaluate: Wav2Vec2 [9], WhisperX [10] (using the whisper v2 large as base model), fine-tuned XLSR-53 Wav2Vec2 [30], HuBERT [34], Seamless M4T v2 [48], and Whisper v3 [1].

Considering the different existing ways to (2) generate a representation for acoustic information, we selected 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 allows the extraction of two levels of information: low-level descriptors and functionals. We use the two levels to compare 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 Transformer library [2] to generate the embeddings: MiniLM-L12, mpnet, and MiniLM-L3. We selected them based on the speed reported in the documentation.

We evaluate experiments (2) and (3) using an LSTM network that predicts valence, arousal and dominance. LSTM is a learning model designed to work with sequential data, which fits the scenario of our experiments. We based our network architecture on a previous work from Atmaja and Akagi [6]. Figure 5.2 illustrates the architecture.

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 created 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 , 100
Activation Function on LSTM	linear, tanh
Output Activation Function	linear, tanh

Table 5.1 – LSTM experimental configuration set

[1] https://huggingface.co/openai/whisper-large-v3
[2] https://www.sbert.net/

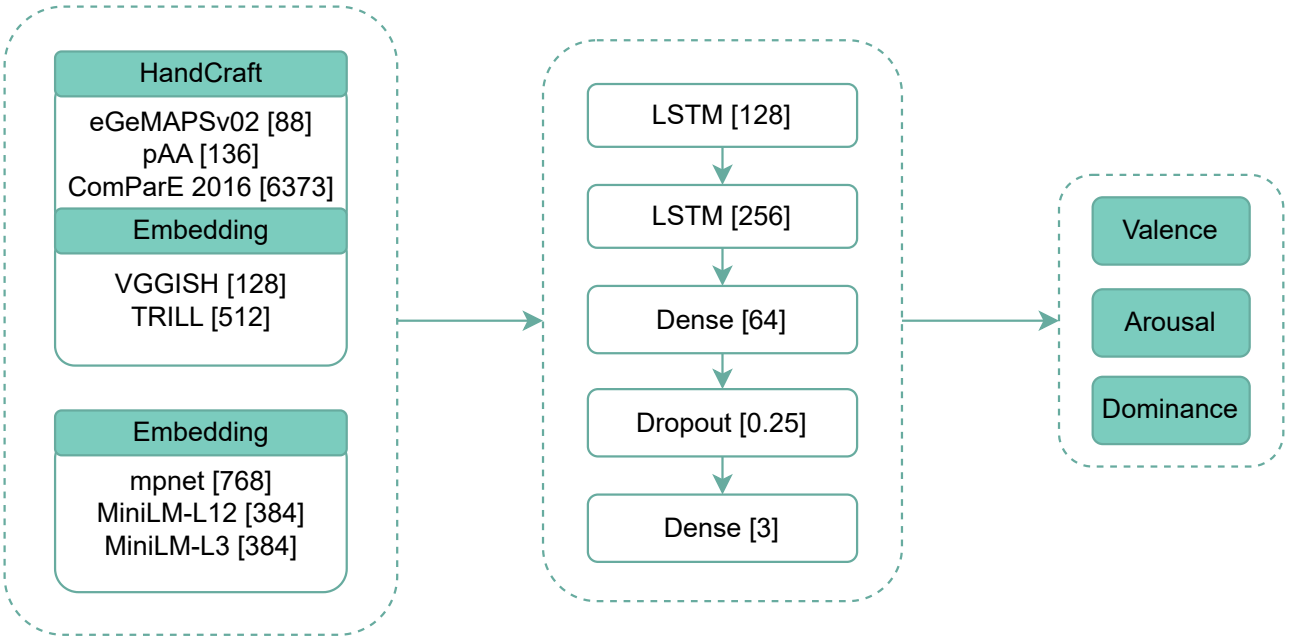

Figure 5.2 – LSTM architecture for acoustic and text features

Our LSTM architecture was implemented using the Keras framework [3]. The final parametrization, considering all the possibilities based on the parameters in Table 5.1, 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. We 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 2.5). The input size is based on the feature dimension, represented in the first column of Figure 5.2.

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 [4] preprocessing to standardize the features. The standard calculation for an x feature is represented in Equation 5.2. Where we have the subtraction of the mean and the division by the standard deviation. In this way, we adjust the distribution of the feature. The evaluation of the MSE and CCC was made through the prediction function from Keras.

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

[3] https://keras.io/
[4] https://scikit-learn.org/

5.2.1 Results

In this section, we will discuss the findings of each experiment. In ASR, the Whisper v3 model achieved the lowest WER, with 0.2262. Table 5.2 shows the complete results for each model. Wav2Vec2 performed better, with 1.33s. However, considering WER, it is a big difference from Whisper v3, as Wav2Vec2 achieves 0.9881. When considering the best choice for our scenario, WhisperX is the best option, considering the second-lowest WER, 0.2738, and the second-highest processing time, 2.803s.

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

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 5.3. 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 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	54.31
ComParE	0.0439 / 0.2028	0.5679 / 0.0764	0.5658 / 0.0764	526.5632	59.10
eGeMAPS LLD	0.0108 / 0.2045	0.0792 / 0.1159	0.0789 / 0.1159	509.2656	57.17
eGeMAPS	**0.2052** / 0.1819	**0.6066** / 0.0704	**0.6086** / 0.0709	509.2656	57.17
pAA	0.136 / 0.1923	0.5813 / 0.075	0.5803 / 0.075	81.357	9.13
TRILL	0.1978 / 0.1887	0.5308 / 0.0769	0.5308 / 0.0769	1099.2904	123.40
VGGISH	0.1751 / 0.1932	0.5694 / 0.0751	0.5694 / 0.0751	**26.4742**	**2.97**
Text Evaluation					
MiniLM-L12	0.1292 / 0.1952	0.0917 / 0.1192	0.0913 / 0.1192	11.9568	1.34
mpnet	0.0412 / 0.2022	0.0245 / 0.1202	0.0226 / 0.1193	11.8198	1.32
MiniLM-L3	**0.3238** / 0.1875	**0.2057** / 0.1125	**0.2057** / 0.1125	**4.1988**	**0.47**

Table 5.3 – Acoustic features results

In experiment (2), the processing is around six times faster than the one for audio. As we can see in Table 5.3, 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 an ablation study to get better results using these representations.

5.3 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.* [8]. We consider the (1) model level, (2) feature level, (3) decision-level fusion, and (4) average from acoustic and linguistic features.

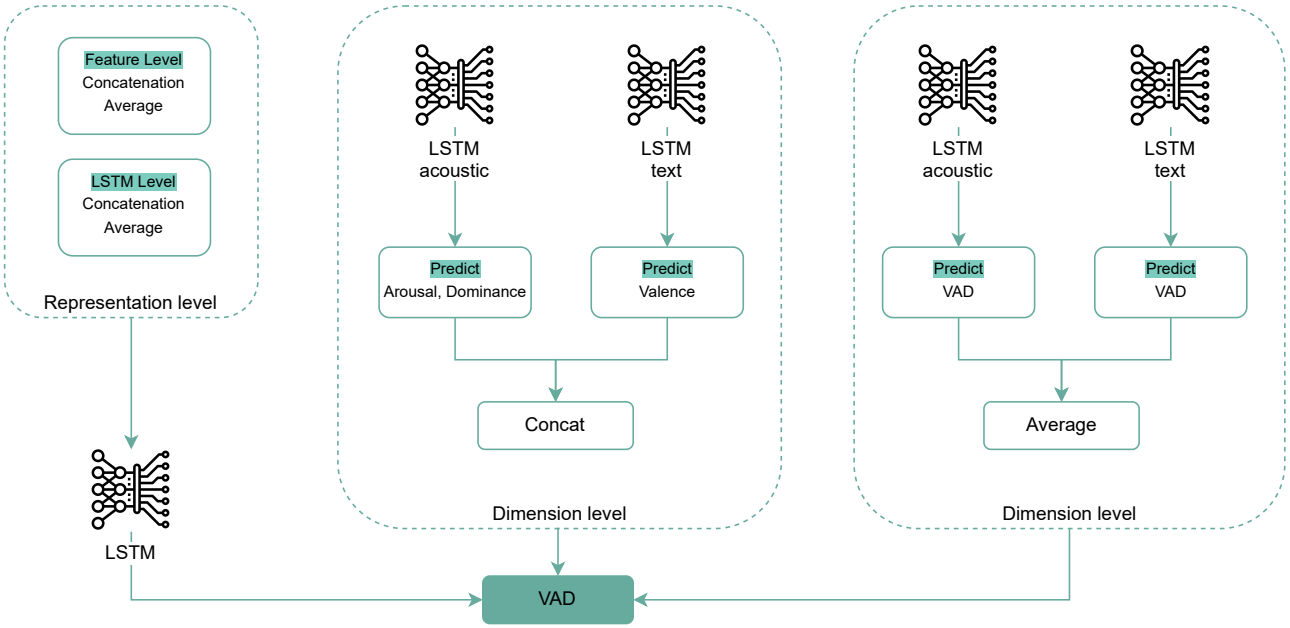

Figure 5.3 – Different structures for fusion concatenation

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 Keras layer, keeping both sentence and audio embedding with the same dimension was necessary. In this case, we use two approaches:

reduce dimensionality from sentence embedding or standardize each utterance to 3 seconds, followed by a flatten function to VGGIsh output.

To reduce the dimensionality of the sentence embedding, the PCA matrix decomposition from Sklearn was applied, reducing it from 384 to 128. At the feature level, we used the Numpy Concat function to concatenate both features. For the average, we first applied the same PCA function; after that, we used the Numpy Average function. To keep the original size of the sentence embeddings, we tried to pad or trim each utterance to 3 seconds and apply a flatten function to transform the VGGIsh output, which contains one vector per second of audio.

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 of an LSTM with two outputs: arousal and dominance dimensions. 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.

5.3.1 Results

We verified that optimizing some parameters of our LSTM produces better results for unimodal approaches, but this is not our focus here. An interesting behavior 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**
Unimodal - baseline			
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
Dimension Level			
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
Concat (VGGIsh flatten + MiniLM-L3) VAD	0.3219 / 0.2012	0.4109 / 0.1074	0.4109 / 0.1074
Average (VGGIsh flatten + MiniLM-L3) VAD	0.029 / 0.2022	0.3446 / 0.0954	0.3442 / 0.0957

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. However, the results are worse than when compared with the unimodal features. Comparing each dimension, valence achieves lower results than only MiniLM, while arousal and dominance are lower than with VGGIsh.

The best results were obtained using the concatenation with PCA on sentence embeddings at the LSTM level as a Keras layer. We have increased arousal and dominance CCC scores, achieving 0.5915 and 0.5899, respectively. Flattened VGGIsh brings better results only on the valence dimension when compared to the PCA one. This was expected because we kept all the information in the sentence here. We also tested the order of features in concatenation, and the best option is to use VGGIsh first. 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.

5.4 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.

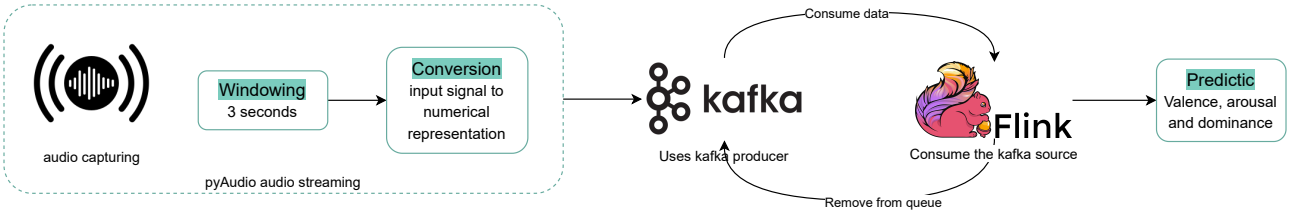

Figure 5.4 – Architecture used for streaming speech emotion recognition

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.

5.5 Discussion

In contradiction to the literature, our highest gain using a bimodal approach, was on the dominance dimension, and not in valence as presented in the related works. We achieved a 17.98% gain in CCC in comparison to the Atmaja and Akagi [6] approach. This is more correlated to the way used to represent the audio withVGGIsh. The incorporation of sentence embeddings adds only 4.12% of CCC.

When we compare the results for valence using only the Mini LM L3 model, they are similar to the bimodal approach of Atmaja and Akagi [6] (0.418 vs 0.4165). Our main issue is the dimensionality reduction for using sentence embedding in the concatenation layer in the Keras model. This is necessary to obtain a sentence embedding with the same number of dimensions as the audio embedding. In the case where we only use PCA as input, CCC is reduced to 0.1055. The concatenation provided a better result, with 0.1431 CCC. But it is still worse than the original size one. This occurs because we apply the PCA after generating the embedding. A possible solution is adding a new dense layer to the Mini LM model and producing the embedding directly with 128 dimensions.

From the most recent machine learning approaches to extract information from audio, we evaluate the VGGIsh and TRILL models, regarding that they are used to feed our LSTM network. Another possible option is to use a CNN network with features from

Wav2Vec2, Wav2Vec2-BERT 2.0, Hubert, and another model that generates more complex representations. Wav2Vec2-BERT 2.0, for example, creates a representation with 1024 dimensions for each *x* ms. To be able to use only one dimension, we apply an average function to the VGGIsh matrix embedding. They produce an array for each second of audio input.

Recent reviews like Geetha *et al.* [26] and Lieskovská *et al.* [43], show a direction for future works in real-world applications that can be used in real-time. To make this possible, the processing time must be considered. However, current publications did not show the processing time necessary to execute their approach. The main focus is the feature selection for better results and the model's architecture. With the LSTM, the total prediction time for our test set was 1.2794 seconds.

Wundt and Judd [86] define that depending on the symptomatic nature of emotions, one of the forms of expressive movements is the expression of ideas. Which can be pantomimetic or descriptive. Due to genetic relationships with speech, it has a special psychological meaning. So, due to the importance of expressing ideas in emotion expression and the lack of diverse and large datasets [26], sentence representations add contextual information to predict the valence and give a modest contribution to the arousal and dominance dimension. The sentence embeddings are the best options when considering the sentence's meaning. The results on valence when using only the Mini LM L3 reflect the good results on the sentiment evaluation databases (see Section 2).

It is controversial to consider that speech emotion recognition can be done in real-time. This is because when we consider the use of sentence embedding, the sentence must be complete to gain more context and meaning. Even if we use real-time transcription, we will deal with, in the better case, words. So, considering the average length of the annotated data chunks from IEMOCAP and MSP-PODCAST, we determine our windowing time to be 3 seconds of utterances.

5.6 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

This work introduces a dimensional speech emotion recognition approach using bimodal features. Our contribution was given in five main aspects: (1) the identification of the better approach for automatic speech recognition; (2) the identification of the better way to generate the sentence embeddings for SER; (3) the identification of the better option between hand-crafted features and audio embedding for acoustic representation. (4) the identification of better options for feature fusion. (5) an architecture to execute SER on a streaming environment.

To achieve our objective, we split our work into five steps. First, we evaluate some of the state-of-the-art models for (1) automatic speech recognition: HuBERT, Wav2Vec2, WhisperX, Whisper V3, Wav2vec2 Large xlsr, and Seamless M4T v2; (2) hand-crafted features and audio embedding for generate acoustic features: ComParE, eGeMAPS, and pAA feature sets and TRILL and VGGIsh embeddings; (3) sentence embeddings models for text representations: MiniLM L12, MiniLM L3, and mpnet.

We evaluate the best way to use acoustic and textual representations. We define WhisperX as the automatic speech recognition model, VGGIsh, and MiniLM L3 for acoustic and textual representation. We explored fusion at the representation level, creating concatenation and average representations at the feature and LSTM levels, adding a new layer to our LSTM network. We also explore the concatenation and average for dimension level but use distinct approaches to predict arousal, valence, and dominance scores. Finally, after defining the best way to use the representations, we build our final architecture and empirically explore the parameters of our network. As a result, we achieve 0.5915 of CCC for arousal, 0.4165 for valence, and 0.5899 for dominance.

With the architecture defined and the LSTM model trained, we build a streaming environment to run our pipeline. The final algorithm captures the microphone input in streaming and sends the representation to a Kafka queue every three seconds. The processing occurs in Flink, which will call a request from an external API that returns the predicted AVD values for that utterance.

This research was also accepted for publication in the XXIV Brazilian Symposium on Computing Applied to Health (SBCAS) [31]. In future work, we plan to use a pre-train version of the Mini LM L3 model to directly produce a vector with 128 dimensions as the output. This will increase the CCC for the valence dimension. By consolidating the best features, we also aim to test with new models, such as Transformer, and use different datasets to train and evaluate our approach. Finally, considering the streaming scenario, we aim to add a sink operation and use a visual approach to understand the model prediction output.

REFERENCES

[1] Ahn, J.; Gobron, S.; Silvestre, Q.; Thalmann, D. “Asymmetrical facial expressions based on an advanced interpretation of two-dimensional russells emotional model”. In: proceedings of ENGAGE, 2010, pp. 12.

[2] Akidau, T.; Chernyak, S.; Lax, R. “Streaming systems: the what, where, when, and how of large-scale data processing”. Sebastopol, CA: O’Reilly, 2018, first edition ed., 39-45p.

[3] Alharbi, S.; Alrazgan, M.; Alrashed, A.; Alnomasi, T.; Almojel, R.; Alharbi, R.; Alharbi, S.; Alturki, S.; Alshehri, F.; Almojil, M. “Automatic speech recognition: Systematic literature review”, *IEEE Access*, vol. 9, September 2021, pp. 131858–131876.

[4] Andrade, H. C. M.; Gedik, B.; Turaga, D. S. “Fundamentals of Stream Processing: Application Design, Systems, and Analytics”. USA: Cambridge University Press, 2014, 1st ed., 45-51p.

[5] Association, A. P. “Diagnostic and Statistical Manual of Mental Disorders: DSM-5-TR”. USA: American Psychiatric Association Publishing, 2022, 56–68p.

[6] Atmaja, B. T.; Akagi, M. “Dimensional speech emotion recognition from speech features and word embeddings by using multitask learning”, *APSIPA Transactions on Signal and Information Processing*, vol. 9, May 2020, pp. e17.

[7] Atmaja, B. T.; Akagi, M. “Two-stage dimensional emotion recognition by fusing predictions of acoustic and text networks using svm”, *Speech Communication*, vol. 126, February 2021, pp. 9–21.

[8] Atmaja, B. T.; Sasou, A.; Akagi, M. “Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion”, *Speech Communication*, vol. 140, May 2022, pp. 11–28.

[9] Baevski, A.; Zhou, H.; Mohamed, A.; Auli, M. “wav2vec 2.0: a framework for self-supervised learning of speech representations”. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 12.

[10] Bain, M.; Huh, J.; Han, T.; Zisserman, A. “Whisperx: Time-accurate speech transcription of long-form audio”. In: Proc. INTERSPEECH, 2023, pp. 4489–4493.

[11] Bertero, D.; Siddique, F. B.; Wu, C.-S.; Wan, Y.; Chan, R. H. Y.; Fung, P. “Real-time speech emotion and sentiment recognition for interactive dialogue systems”. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1042–1047.

[12] Bhangale, K. B.; Kothandaraman, M. “Survey of Deep Learning Paradigms for Speech Processing”, *Wireless Personal Communications*, vol. 125, July 2022, pp. 1913–1949.

[13] Boehner, K.; DePaula, R.; Dourish, P.; Sengers, P. “Affect: From information to interaction”. In: Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility, 2005, pp. 59–68.

[14] Brunet, K.; Taam, K.; Cherrier, E.; Faye, N.; Rosenberger, C. “Speaker Recognition for Mobile User Authentication: An Android Solution”. In: 8ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d’Information (SAR SSI), 2013, pp. 10.

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

[16] Conneau, A.; Kiela, D. “SentEval: An evaluation toolkit for universal sentence representations”. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, 2018, pp. 1269.

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

[18] de Lope, J.; Graña, M. “An ongoing review of speech emotion recognition”, *Neurocomputing*, vol. 528, April 2023, pp. 1–11.

[19] Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. “BERT: pre-training of deep bidirectional transformers for language understanding”. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

[20] Ding, S.; Chen, T.; Gong, X.; Zha, W.; Wang, Z. “Autospeech: Neural architecture search for speaker recognition”. Source: https://arxiv.org/abs/2005.03215, October 2023.

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

[22] Ekman, P. “Basic Emotions”. John Wiley and Sons, Ltd, 1999, chap. 3, pp. 45–60.

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

[24] Eyben, F.; Wöllmer, M.; Schuller, B. “Opensmile: The munich versatile and fast open-source audio feature extractor”. In: Proceedings of the 18th ACM International Conference on Multimedia, 2010, pp. 1459–1462.

[25] Friedman, E.; Tzoumas, K. “Introduction to Apache Flink: Stream Processing for Real Time and Beyond”. Sebastopol, CA: O’Reilly Media, Inc., 2016, 1st ed., 21p.

[26] Geetha, A.; Mala, T.; Priyanka, D.; Uma, E. “Multimodal emotion recognition with deep learning: Advancements, challenges, and future directions”, *Information Fusion*, vol. 105, March 2024, pp. 102–218.

[27] Ghriss, A.; Yang, B.; Rozgic, V.; Shriberg, E.; Wang, C. “Sentiment-aware automatic speech recognition pre-training for enhanced speech emotion recognition”. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2022, pp. 7347–7351.

[28] Giannakopoulos, T. “pyaudioanalysis: An open-source python library for audio signal analysis”, *PLOS ONE*, vol. 10, 12 2015, pp. 1–17.

[29] Goodfellow, I.; Bengio, Y.; Courville, A. “Deep Learning”. Cambridge, Massachusetts: MIT Press, 2016, 453p.

[30] Grosman, J. “Fine-tuned XLSR-53 large model for speech recognition in English”. Source: https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english, December 2023.

[31] Guder, L.; Aires, J. P.; Meneguzzi, F.; Griebler, D. “Dimensional Speech Emotion Recognition from Bimodal Features”. In: Brazilian Symposium on Computing Applied to Health, 2024, pp. 12.

[32] 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.; Wilson, K. “CNN architectures for large-scale audio classification”. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 131–135.

[33] Hochreiter, S.; Schmidhuber, J. “Long short-term memory”, *Neural Comput.*, vol. 9, November 1997, pp. 1735–1780.

[34] Hsu, W.; Bolte, B.; Tsai, Y. H.; Lakhotia, K.; Salakhutdinov, R.; Mohamed, A. “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”. 2106.07447, Source: https://arxiv.org/abs/2106.07447, December 2023.

[35] Huang, X.; Acero, A.; Hon, H.-W. “Spoken language processing: a guide to theory, algorithm, and system development”. Upper Saddle River, NJ: Prentice Hall PTR, 2001, 26–27p.

[36] Ispas, A.-R.; Deschamps-Berger, T.; Devillers, L. “A multi-task, multi-modal approach for predicting categorical and dimensional emotions”. In: ACM International Conference Proceeding Series, 2023, pp. 311 – 317.

[37] Julião, M.; Abad, A.; Moniz, H. “Exploring text and audio embeddings for multi-dimension elderly emotion recognition”. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2020, pp. 2067–2071.

[38] Koh, E. S.; Dubnov, S. “Comparison and analysis of deep audio embeddings for music emotion recognition”. Source: https://arxiv.org/abs/2104.06517, December 2023.

[39] Koolagudi, S. G.; Murthy, Y. V. S.; Bhaskar, S. P. “Choice of a classifier, based on properties of a dataset: case study-speech emotion recognition”, *International Journal of Speech Technology*, vol. 21, March 2018, pp. 167–183.

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

[41] Lech, M.; Stolar, M.; Best, C.; Bolia, R. “Real-time speech emotion recognition using a pre-trained image classification network: Effects of bandwidth reduction and companding”, *Frontiers in Computer Science*, vol. 2, May 2020, pp. 14.

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

[43] Lieskovská, E.; Jakubec, M.; Jarina, R.; Chmulík, M. “A review on speech emotion recognition using deep learning and attention mechanism”, *Electronics*, vol. 10, January 2021, pp. 1163.

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

[45] MacAry, M.; Tahon, M.; Esteve, Y.; Rousseau, A. “On the use of self-supervised pre-trained acoustic and linguistic features for continuous speech emotion recognition”. In: IEEE Spoken Language Technology Workshop, 2021, pp. 373–380.

[46] Malik, M.; Malik, M. K.; Mehmood, K.; Makhdoom, I. “Automatic speech recognition: a survey”, *Multimedia Tools and Applications*, vol. 80, March 2021, pp. 9411–9457.

[47] Mehrabian, A. “Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament”, *Current Psychology*, vol. 14, December 1996, pp. 261–292.

[48] Meta AI. “Seamlessm4t: Massively multilingual and multimodal machine translation”. 2308.11596, Source: https://arxiv.org/abs/2308.11596, December, 2023.

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

[50] Nagrani, A.; Chung, J. S.; Zisserman, A. “VoxCeleb: A large-scale speaker identification dataset”. In: Interspeech, 2017, pp. 2616–2620.

[51] Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A. Y. “Multimodal deep learning”. In: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 689–696.

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

[53] Perronnin, F.; Dance, C. “Fisher kernels on visual vocabularies for image categorization”. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[54] Pham, N. T.; Dang, D. N. M.; Pham, B. N. H.; Nguyen, S. D. “SERVER: Multi-modal Speech Emotion Recognition using transformer-based and vision-based embeddings”. In: Proceedings of the 8th International Conference on Intelligent Information Technology, 2023, pp. 234–238.

[55] Picard, R. W. “Affective Computing”. Cambridge, MA: MIT Press, 1997, 292p.

[56] Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. “Robust speech recognition via large-scale weak supervision”. In: Proceedings of the 40th International Conference on Machine Learning, 2023, pp. 27.

[57] Ravanelli, M.; Zhong, J.; Pascual, S.; Swietojanski, P.; Monteiro, J.; Trmal, J.; Bengio, Y. “Multi-task self-supervised learning for robust speech recognition”. Source: https://arxiv.org/abs/2001.09239, October 2023.

[58] Reimers, N.; Gurevych, I. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019, pp. 3980–3990.

[59] Roberts, K.; Roach, M. A.; Johnson, J.; Guthrie, J.; Harabagiu, S. M. “EmpaTweet: Annotating and detecting emotions on Twitter”. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12), 2012, pp. 3806–3813.

[60] Roger, V.; Farinas, J.; Pinquier, J. “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, August 2022, pp. 19.

[61] Russell, J. “A circumplex model of affect”, *Journal of personality and social psychology*, vol. 39, December 1980, pp. 1161–1178.

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

[63] Scherer, K. R. “What are emotions? and how can they be measured?”, *Social Science Information*, vol. 44, December 2005, pp. 695–729.

[64] Schuller, B.; Steidl, S.; Batliner, A.; Hirschberg, J.; Burgoon, J. K.; Baird, A.; Elkins, A.; Zhang, Y.; Coutinho, E.; Evanini, K. “The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity and native language”. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2016, pp. 2001 – 2005.

[65] Shor, J.; Jansen, A.; Maor, R.; Lang, O.; Tuval, O.; de Chaumont Quitry, F.; Tagliasacchi, M.; Shavitt, I.; Emanuel, D.; Haviv, Y. “Towards learning a universal non-semantic representation of speech”. In: Interspeech, 2020, pp. 140–144.

[66] Simonyan, K.; Zisserman, A. “Very deep convolutional networks for large-scale image recognition”. Source: https://arxiv.org/abs/1409.1556, November 2023.

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

[68] Singh, Y. B.; Goel, S. “A systematic literature review of speech emotion recognition approaches”, *Neurocomputing*, vol. 492, July 2022, pp. 245–263.

[69] Smith, S. W. “The Scientist and Engineer’s Guide to Digital Signal Processing”. San Diego, CA: California Technical Publishing, 1997, 36p.

[70] Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. “X-Vectors: Robust DNN Embeddings for Speaker Recognition”. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2018, pp. 5329 – 5333.

[71] Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; Potts, C. “Recursive deep models for semantic compositionality over a Sentiment Treebank”. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.

[72] Sogancioglu, G.; Verkholyak, O.; Kaya, H.; Fedotov, D.; Cadée, T.; Salah, A.; Karpov, A. “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, 2020, pp. 2097–2101.

[73] Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.-Y. “MPNet: Masked and Permuted Pre-Training for Language Understanding”. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 11.

[74] Srinivasan, S.; Huang, Z.; Kirchhoff, K. “Representation learning through cross-modal conditional teacher-student training for speech emotion recognition”. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 4298–4302.

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

[76] Sun, L.; Lian, Z.; Tao, J.; Liu, B.; Niu, M. “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, 2020, pp. 27–34.

[77] Testa, B.; Xiao, Y.; Sharma, H.; Gump, A.; Salekin, A. “Privacy against real-time speech emotion detection via acoustic adversarial evasion of machine learning”, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, September 2023, pp. 30.

[78] Triantafyllopoulos, A.; Reichel, U.; Liu, S.; Huber, S.; Eyben, F.; Schuller, B. W. “Multistage linguistic conditioning of convolutional layers for speech emotion recognition”, *Frontiers in Computer Science*, vol. 5, February 2023, pp. 1072479.

[79] Triantafyllopoulos, A.; Wagner, J.; Wierstorf, H.; Schmitt, M.; Reichel, U.; Eyben, F.; Burkhardt, F.; Schuller, B. “Probing speech emotion recognition transformers for linguistic knowledge”. In: Proc. Interspeech, 2022, pp. 146–150.

[80] Van Houdt, G.; Mosquera, C.; Nápoles, G. “A review on the long short-term memory model”, *Artificial Intelligence Review*, vol. 53, December 2020, pp. 5929–5955.

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

[82] Wang, C.; Ren, Y.; Zhang, N.; Cui, F.; Luo, S. “Speech emotion recognition based on multi-feature and multi-lingual fusion”, *Multimedia Tools and Applications*, vol. 81, February 2022, pp. 4897–4907.

[83] Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; Zhou, M. “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, 2020, pp. 13.

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

[85] Williams, C. E.; Stevens, K. N. “Emotions and Speech: Some Acoustical Correlates”, *The Journal of the Acoustical Society of America*, vol. 52, October 1972, pp. 1238–1250.

[86] Wundt, W.; Judd, C. “Outlines of Psychology”. W. Engelmann, 1897, 173p.

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