
F3

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's Thesis

# Emotion Detection from Speech and Written Text

**Petr Stádník**
Artificial Intelligence and Computer Science,
Open Informatics

May 2024
Supervisor: doc. Ing. Daniel Novák, Ph.D.

# BACHELOR‘S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Stádník Petr** Personal ID number: **498942**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

## II. Bachelor’s thesis details

Bachelor’s thesis title in English:

**Emotion Detection from Speech and Written Text**

Bachelor’s thesis title in Czech:

**Detekce emocí z řeči a psaného textu**

Guidelines:

1. Get an overview of emotion recognition methods in speech and text and explore the basic libraries available.
2. Gain an overview of datasets that are used to train models for emotion detection in speech. Search and analyze both English and Czech datasets.
3. Implement a complete process chain using the selected libraries or models. The process chain should detect emotions in both Czech and English spoken speech.

Bibliography / sources:

[1] H. Zou, Y. Si, C. Chen, D. Rajan and E. S. Chng, "Speech Emotion Recognition with Co-Attention Based Multi-Level Acoustic Information," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022
[2] J. Ye, X. -C. Wen, Y. Wei, Y. Xu, K. Liu and H. Shan, "Temporal Modeling Matters: A Novel Temporal Emotional Modeling Approach for Speech Emotion Recognition," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023
[3] Ziyang Ma, Zhisheng Zheng, Jiaxin Ye, Jinchao Li, Zhifu Gao, Shiliang Zhang, Xie Chen, emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation, https://arxiv.org/abs/2312.15185

Name and workplace of bachelor’s thesis supervisor:

**doc. Ing. Daniel Novák, Ph.D. Analysis and Interpretation of Biomedical Data FEE**

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **23.01.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

| doc. Ing. Daniel Novák, Ph.D. | prof. Dr. Ing. Jan Kybic | prof. Mgr. Petr Páta, Ph.D. |
|---|---|---|
| Supervisor’s signature | Head of department’s signature | Dean’s signature |

## III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

| . | |
|---|---|
| Date of assignment receipt | Student’s signature |

# Acknowledgement / Declaration

I would like to thank doc. Ing. Daniel Novák, Ph.D. for the time he spent guiding my work, his advice, and his patience. I would also like to thank the other members of our research team for their valuable tips and the pleasant atmosphere we had during the numerous meetings. These are: Klára Losenická, Štěpán Bořek, Cheng Kang, MSc., Bc. Fabián Bodnár, Bc. Petr Karlík, and Varvara Shorina.

In the end, I would like to thank my family and friends for their important support during the whole process.

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague 23. 5. 2024

Petr Stádník

# Abstrakt / Abstract

V této práci se věnujeme emocím a jejich rozpoznávání primárně z řeči, ale představujeme také model pro rozpoznávání emocí z textu. Začínáme tím, že si představíme emoce jako takové a různé emoční teorie. Poté si ukážeme, jaké vlastnosti záznamu řeči jsou pro nás při rozpoznávání klíčové a jak je z řečové nahrávky získáme. Dáváme nahlédnout tomu, jak se vyvíjelo rozpoznávání emocí z řeči a jaké metody k tomu byly používány dříve a jaké jsou dnes. V další kapitole představujeme několik vybraných anglických a českých datasetů. Jádrem naší práce je představení několika existujících moderních modelů pro rozpoznávání emocí z řeči a jejich otestování na již zmíněných datasetech. Výsledky jsme porovnali a vybrali ten nejlepší model, kterým je emotion2vec, s kterým jsme poté provedli další testy s přidáním a redukcí šumu. K tomuto modelu poté přidáváme ještě model pro rozpoznání emocí z textu a pro oba jsme pak zpracovali a naprogramovali algoritmus, jehož vstupem může být buď samostatně záznam řeči nebo text, nebo kombinace obojího a výstupem jsou obsažené emoce. Došli jsme k závěru, že kombinací výsledků z obou modelů můžeme eliminovat nesprávně klasifikované emoce, ale také ztrácíme mnoho správně klasifikovaných emocí, které jsou nyní klasifikovány jako neznámé.

**Klíčová slova:** rozpoznávání emocí z řeči, rozpoznávání emocí z textu, SER, wav2vec, huBert, emotion2vec, emoce, sentiment

In this work, we are focusing on emotions and their recognition primarily from speech, but we are also introducing a model for recognizing emotions from text. We begin by introducing emotions and various emotion theories. We are describing which features of the speech recording are the most important for emotion recognition and how we can get them from the speech recording. We are introducing how the recognition of emotions from speech has been developing in the last decades which methods were used for this before and what methods are used nowadays. In the next part, we are introducing several selected English and Czech datasets. The core of our work is presenting several existing state-of-the-art models for recognizing emotions from speech and their testing on the previously mentioned datasets. After we had compared the results we selected the best model which is emotin2vec and we tested it with the addition and reduction of noise. Finally, we add a model for recognizing emotions from text, and for both models, we have written an algorithm, the input of which is either a speech recording or text, or a combination of both, and the output is the contained emotion. In the end, have tested the functionality of this algorithm. We have concluded that by combining results from both models we can eliminate incorrectly classified emotions as emotions from opposite sentiment, but we lose many correctly classified emotions, which are now classified as unknown.

**Keywords:** speech emotion recognition, text emotion recogition, SER, wav2vec, huBert, emotion2vec, emotios, sentiment

# Contents /

# Tables / Figures

# Chapter 1
# Introduction

Human emotions were an area that was for a long time for computers hidden. But nowadays it is changing and many researchers across the world are on the way to teach computers to recognize human emotions and react according to them.

Speech emotion recognition can be useful and applicable in many areas from call centers to medical emergencies, security, and many others. In this work, we try to develop a process chain that will be hopefully later implemented into software, just emerging at Czech Technical University, for Národní linka pro odvykání (National Quitting Line).

This work is mostly focused on emotion recognition from speech and slightly on emotion recognition from text. For this purpose, we will first briefly take a look at what in fact emotions are, how they were developed, and introduce a few most common emotion theories, to better understand what we actually want to recognize and how to correctly understand results.

After that, we will focus on state-of-the-art knowledge in this area. We will find, introduce, and analyze a few selected English and Czech datasets which we will then use for testing. Along with them we will also search for models used for speech emotion recognition tasks, introduce them, and, what will be the most important part, test them on those datasets to pick the best model on which we will perform further tests including testing their behavior with noise reduction and white noise added.

Since the goal of this work is to prepare a process chain that is supposed to recognize emotion from the text input only as same as from speech input only and finally from both combined, we will take the best model for speech emotion recognition and the best model for text emotion recognition and prepare process chain to maximize correctness of the output for all these three scenarios.

Code implementation will be published on GitHub.

# Chapter 2
# Brief history of emotions

Emotions have accompanied us since ancient times. We meet them every day and take them for granted. Charles Darwin, the founder of evolutionary biology, already dealt with emotions in his book The Expression of the Emotions in Man and Animals published in 1872 [1]. A Few pictures with original descriptions of animals expressing emotions from this historical book you can see below.

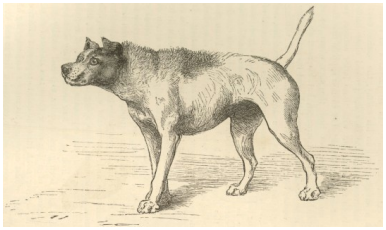

**Figure 2.1.** Dog approaching another dog with hostile intentions.

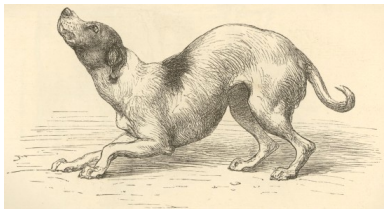

**Figure 2.2.** The Same in a humble and affectionate frame of mind.

It is interesting, that emotions are so natural for people that basic emotions are understood by people all over the world on all continents, even though they speak different languages and have developed completely different cultures.

Of course, we can find there small differences. For example, it is known [2] that people from Eastern Asia focus much more on the eyes area while recognizing emotions from a face. Compared to it, for Western people like Europeans, Americans, etc. is more important mouth area. It has interesting consequences. One of them, which we can meet in a time of modern technologies, is differently developed emoticons reflecting this difference. Examples can be seen in the picture below.

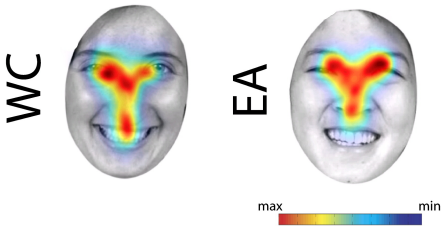


**Figure 2.3.** Areas where people are focusing while recognising emotions in Western (WC) and Eastern Asia (EA).

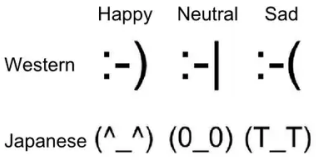


**Figure 2.4.** Difference between emoticons used in Western and Japanese.

Apart from the visual sense, our emotions are also included in our speaking. Not only in the content of our communication but also in the intonation and timbre of the voice. For example, when we see a dog we can write a sentence “There is a dog.” From this text, someone else is not able to distinguish if we are happy, because it is our favorite dog or we are fearful because we are scared of dogs. But by intonation in a speech, there would be a big difference.

Hence there is an idea that emotion recognition from text is not enough. In this work, we will look at how far computer science in this field is. We will try different models and try to produce a reliable process chain capable of recognizing emotions from various given sentences in text or speech form.

## 2.1 What the emotions are?

In history, we can find a variety of emotion theories developed by different psychologists. Most of them declare the existence of 5 or more basic emotions. As a basic emotions are often referred these emotions which are believed to be universal across cultures. In this work, we will present a few emotion theories to represent how various the look on emotions is even in the scientific community.

We will start with probably the most famous theory written by American psychologist professor Paul Ekman [3].

### 2.1.1 Paul Ekman emotion theory

Firstly according to his theory from 1972, there were 6 emotions which we can distinguish [4]. These are:

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

But in his later studies published around 1990, he came up with the seventh emotion. He found enough evidence to distinguish contempt and disgust. So he added contempt as the seventh basic emotion [4].

### 2.1.2 Robert Plutchik emotion theory

Another important name in the area of emotion research is an American psychologist Robert Plutchik [5]. In his theories, he recognized 8 basic emotions in contra-pairs:

- Joy versus sadness
- Anger versus fear
- Trust versus disgust
- Surprise versus anticipation

Also in his psychoevolutionary theory [6] he declared 10 postulates about emotions: (These related to this work have been highlighted)

1. The concept of emotion is applicable to all evolutionary levels and applies to animals as well as to humans.
2. Emotions have an evolutionary history and have evolved various forms of expression in different species.
3. Emotions served an adaptive role in helping organisms deal with key survival issues posed by the environment.
4. **Despite different forms of expression of emotions in different species, there are certain common elements, or prototype patterns, that can be identified.**
5. **There is a small number of basic, primary, or prototype emotions.**

6. **All other emotions are mixed or derivative states; that is, they occur as combinations, mixtures, or compounds of the primary emotions.**
7. **Primary emotions are hypothetical constructs or idealized states whose properties and characteristics can only be inferred from various kinds of evidence.**
8. Primary emotions can be conceptualized in terms of pairs of polar opposites.
9. **All emotions vary in their degree of similarity to one another.**
10. **Each emotion can exist in varying degrees of intensity or levels of arousal.**

He is also well known for creating Plutchik's emotion wheel in 1980 in order to describe how emotions are related.

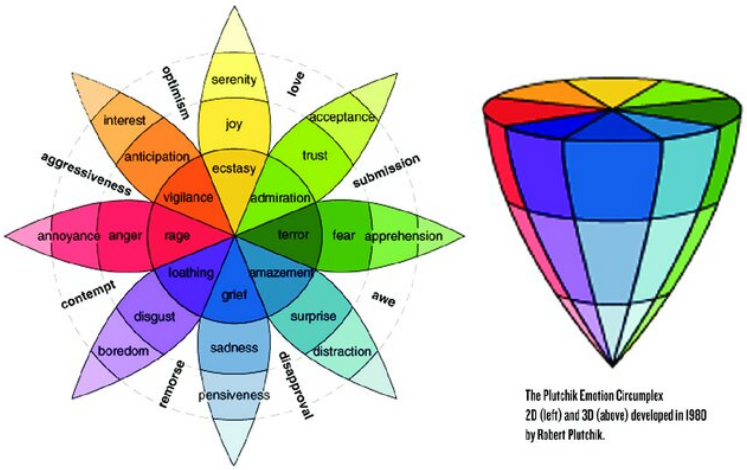


**Figure 2.5.** Plutchik's emotion wheel (left) and the accompanying emotion cone (right).[7]

Emotions located between leaves are supposed to be a combination of emotions around, for example, love is a combination of joy and trust.

### 2.1.3 Many other approaches

It is important to say, that there are many different points of view, on how to think about basic emotions. As declares following table published in a work named What's Basic About Basic Emotions by Andrew Ortony and Terence J. Turner in 1989 [8]. Another problem is that there are often many names for the same emotion.

| Reference | Year | Fundamental (basic) emotions | Basis for inclusion |
|---|---|---|---|
| Arnold | 1960 | anger, aversion, courage, dejection, desire, despair, fear, hate, hope, love, sadness | Relation to action tendencies |
| Ekman, Friesen and Ellsworth | 1982 | anger, disgust, fear, joy, sadness, surprise | Universal facial expressions |
| Frijda | 1986 | desire, happiness, interest, surprise, wonder, sorrow | Forms of action readiness |
| Gray | 1982 | rage and terror, anxiety, joy | Hardwired |
| Izard | 1971 | anger, contempt, disgust, distress, fear, guilt, interest, joy, shame, surprise | Hardwired |
| James | 1884 | fear, grief, love, rage | Bodily involvement |
| McDougall | 1926 | anger, disgust, elation, fear, subjection, tender-emotion, wonder | Relation to instincts |
| Mowrer | 1960 | pain, pleasure | Unlearned emotional states |
| Oatley and Johnson-laird | 1987 | anger, disgust, anxiety, happiness, sadness | Do not require propositional content |
| Panksepp | 1982 | expectancy, fear, rage, panic | Hardwired |
| Plutchik | 1980 | acceptance, anger, anticipation, disgust, joy, fear, sadness, surprise | Relation to adaptive biological processes |
| Tomkins | 1984 | anger, interest, contempt, disgust, distress, fear, joy, shame, surprise | Density of neural firing |
| Watson | 1930 | fear, love, rage | Hardwired |
| Weiner and Graham | 1984 | happiness, sadness | Attribution independent |

**Table 2.1.** Table of basic emotions according different researchers with various appproach what basic emotions are.

# Chapter 3
# Speech and text emotion recognition

Nowadays scientists try different approaches to recognize emotions. One of the most common approaches is recognition using facial expressions from images. Another commonly used way is to recognize emotions from written text. And finally directly from the voice record. The benefits and disadvantages of these methods are compared in the following sections. Common and precious is a combination of voice and facial expression.

## 3.1 Introduction to speech emotion recognition

Already in the early 1990s researchers began exploring how to automatically detect emotions in human speech using machine learning methods.

The first important task is to find some features of speech recording, which are supposed to differ for each emotion, so on which we can base classification.

## 3.2 Key audio features

As well as we recognize emotions from the face, where we are mostly focusing on how the mouth is curved or open and how the eyes area looks, it is important to find some criteria and features in a voice record, that we can use for emotion recognition.

It is believed that prosodic features of the voice signal are the most important indicators of a speaker's emotional state. Researches indicate that fundamental frequency, energy, and formant frequencies are potentially effective parameters to distinguish certain emotional states [9].

The state-of-the-art models and systems for speech recognition and speech emotion recognition commonly use the following features.

### 3.2.1 Fundamental frequency $F_0$

Typical values of fundamental voice frequency $F_0$ for men are 120 Hz and 210 Hz for women. The mean values change slightly with age. For men, the decrease in $F_0$ is most dramatic during puberty [10].

It was shown in many studies, that the $F_0$-range is influenced by various factors such as the language, the type of content, the type of discourse, and finally the emotional state of the speaker [10].

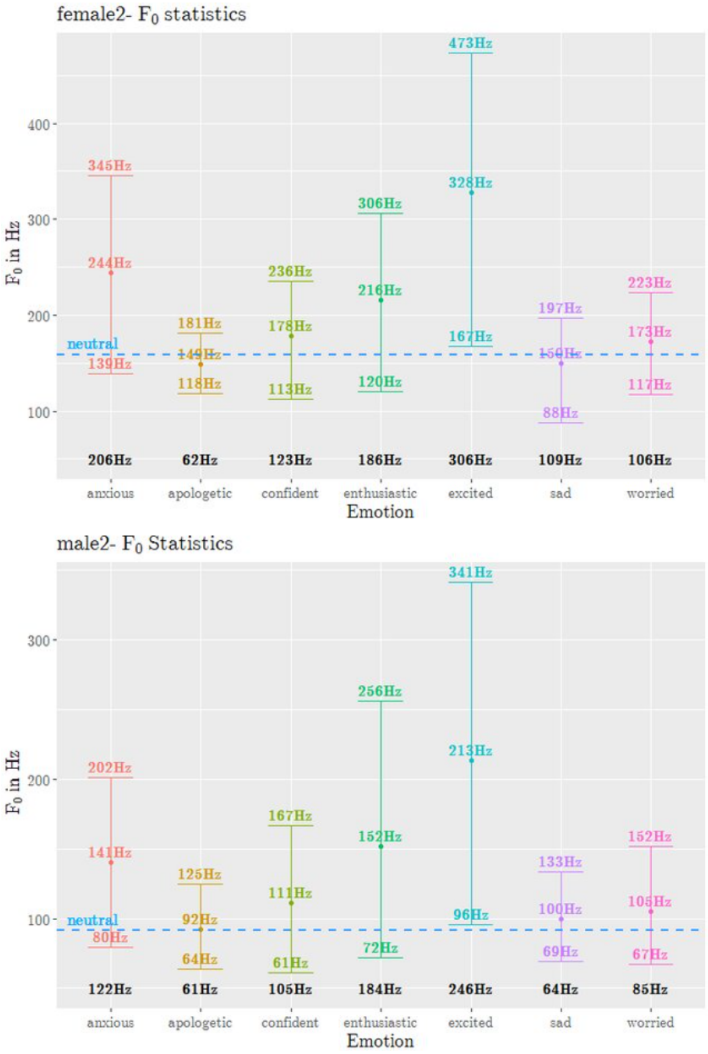


**Figure 3.1.** In these two charts from [11] you can see fundamental frequency statistics of different emotions first for female and then for male speakers. The dot on each line represents the mean of fundamental frequency and the upper and lower bounds indicate the maximum and minimum values, respectively. The bold black number at the bottom is the fundamental frequency range. The dashed line represents the results for the neutral state.

### 3.2.2 Spectogram image

Spectrogram image simply represents which frequencies are contained in the input signal in exact time. Usually, we can see it as a graph with two geometric dimensions x-axis represents time, and the y-axis represents frequency. A color as a third dimension in each point represents the amplitude of a particular frequency at that time.

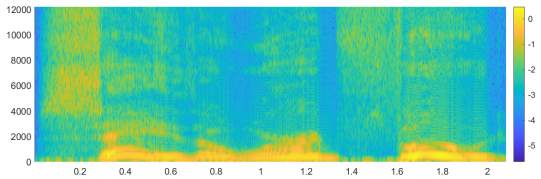


**Figure 3.2.** Spectogram image of recording with happy intonation.

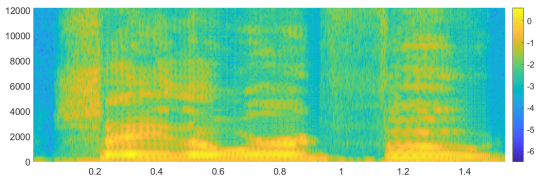


**Figure 3.3.** Spectogram image of recording with the same content but in angry intonation.

Spectrogram images are commonly generated to be used as input images for convolutional neural networks which can be part of the speech emotion recognition process.

### 3.2.3 Mel frequency cepstral coefficients (MFCC)

The main idea of mel frequency cepstral coefficients is to compress vocal information into a small number of coefficients, which can be used for further classification. MFCC is based on known variation of the human ear. MFCC has two types of filters which are spaced linearly at low frequency below 1000 Hz and logarithmic spacing above 1000Hz [12].

Although there is no hard standard for calculating the coefficients from input audio, we will introduce these basics and the most important ones.

1. **Windowing.** First, we need to segment the input file into small overlapping fragments of length around 25ms.
2. **Fourier transform.** Then we convert each fragment, we got from the previous step, from the time domain into the frequency domain.
3. **Mel Filter Bank Processing.** A set of triangular filters is used to compute a weighted sum of filter spectral components so that the output of the process approximates a Mel scale [12].
4. **Discrete Cosine Transform** to convert the log Mel spectrum into the time domain. The set of coefficients is called acoustic vectors. Therefore, each input utterance is transformed into a sequence of acoustic vectors.

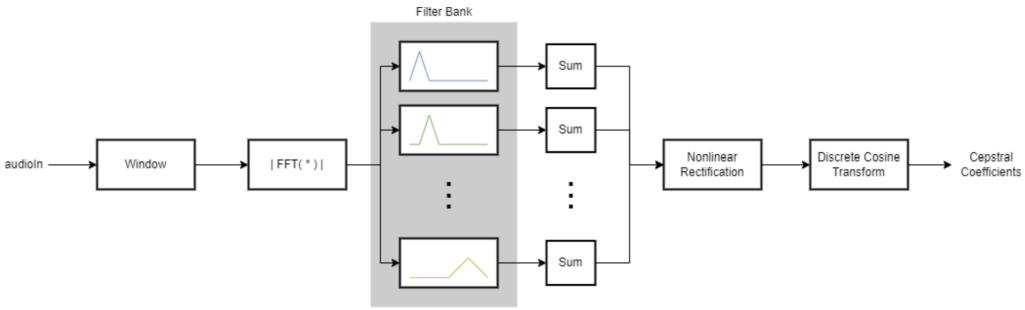


**Figure 3.4.** Diagram describing the creation of Mel frequency cepstrum coefficient step by step. Image from [13].

### 3.2.4 Constant-Q Transform (CQT)

Constant Q transformation transforms the input signal from the time domain into the time-frequency domain so that the center frequencies of the frequency bins are geometrically spaced and their Q-factors (ratios of the center frequencies to bandwidths) of all bins are equal. In effect, this means that the frequency resolution is better for low frequencies and the time resolution is better for high frequencies [14]. In general, the transform is well suited to musical data. As the range of human hearing covers approximately ten octaves from 20 Hz to around 20 kHz, this reduction in output data is significant.

### 3.2.5 Gaussian Mixture Model (GMM)

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

### 3.2.6 Hidden Markov model (HMM)

HMM is a classifier based on a Markov chain containing a finite number of states. Between these states, you can move with given transition probabilities. HMM is trained by sequences of feature vectors that are representative of the input signal. HMM has a long history in speech recognition [9]. In HMM, states are not observable. Estimation of the parameters in an HMM can be also performed using maximum likelihood.

### 3.2.7 Support vector machine (SVM)

SVM is a newer technique for data classification and regression. Its main idea is to transform the original input set to a high-dimensional feature space by using a kernel function, and then achieve optimum classification in this new feature space [9].

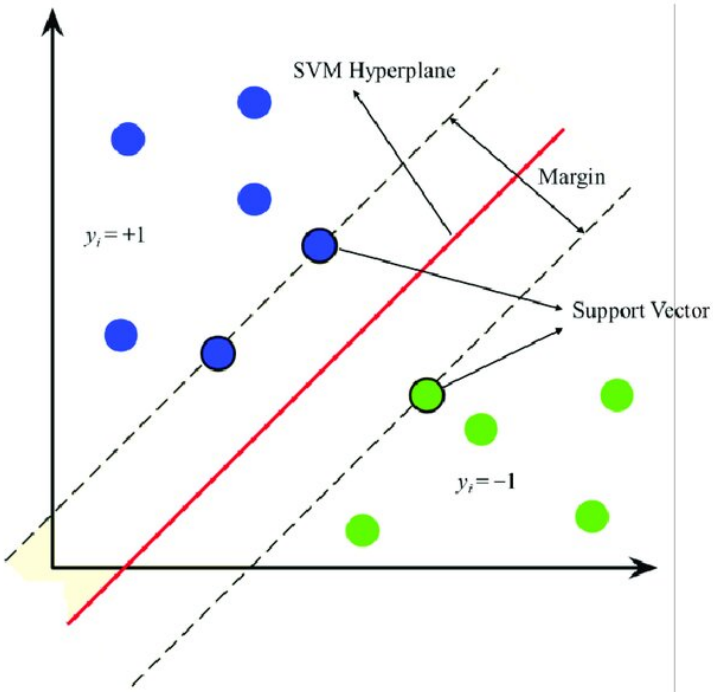


**Figure 3.5.** Example of linear SVM classifier separating two classes [15].

### 3.2.8 K-nearest neighbours (KNN)

KNN algorithm is commonly used for classification and regression tasks. It is based on a simple idea of similarity of the same-labeled data.

KNN algorithm stores the entire training dataset and when some new data point comes for classification, it calculates the distance between the new input data point and all the training examples, using a chosen distance metric such as Euclidean distance.

After that, the algorithm identifies the K nearest points from the training dataset to the new point and classifies the new point as a member of the class which is the most common between K chosen points.

## 3.3 State of the art basic building blocks

Since neural networks become part of this world, researchers are still developing better and better models suitable for wide usage, including language processing area. One of

the top companies developing AI tools for language processing is Meta. Some of their models and tools we will describe below.

Some of these models are also nowadays used as a **feature extractors**. The goal of the feature extractor is to prepare an input of features for the model or classifier. This input is usually an n-dimension vector. In our case, when our input is an audio file, it can consist of extracted audio features listed in the previous section and many others. It can also contain normalization itself.

In this work, we will also use library Transformers developed and published by **Hugging Face platform**. This platform which is nowadays widely used to share models and datasets across the whole machine learning community, including companies developing state-of-the-art models like Meta, Invidia, OpenAI, or Apple, contains around 350,000 models, 75,000 datasets, and 150,000 demo apps publicly available. Also, this platform provides useful tools and libraries. One of them is library **Transformers** [16], which provides for example methods for already mentioned feature extraction.

### 3.3.1 Wav2vec (2.0)

When it comes to model training, there is always a problem with getting enough labeled or transcribed audio data. To deal with this problem, researchers from Meta company are focused on developing systems, which do not need such a large amount of labeled data.

In September 2019 they introduced the wav2vec model, which uses unsupervised pre-training to improve supervised speech recognition. This enables exploiting unlabelled audio data which is much easier to collect than labeled data [17].

This wav2vec model is a convolutional neural network that takes raw audio as input and computes a general representation that can be input to a speech recognition system.

Later, in September 2020 was introduced wav2vec 2.0, which is a framework for self-supervised learning of speech representations from raw audio [18].

Both models were later fine-tuned and used also for emotion segmentation tasks.

### 3.3.2 HuBERT

Self-supervised representation learning for speech recognition, generation, and compression HuBERT was released in June 2021 [19].

Self-supervised approaches for speech representation learning are difficult because of three problems. The first of them is that, there are multiple sound units in each input utterance. The second one is that, there is no lexicon of input sound units during the pre-training phase, and the third one, is that sound units have variable lengths with no explicit segmentation. To deal with these three problems, researchers in Meta proposed the Hidden-Unit BERT (HuBERT) approach for self-supervised speech representation learning. Their approach forces the model to learn a combined acoustic and language model over the continuous inputs. They used a simple k-means algorithm as a teacher as you can see in the diagram below.

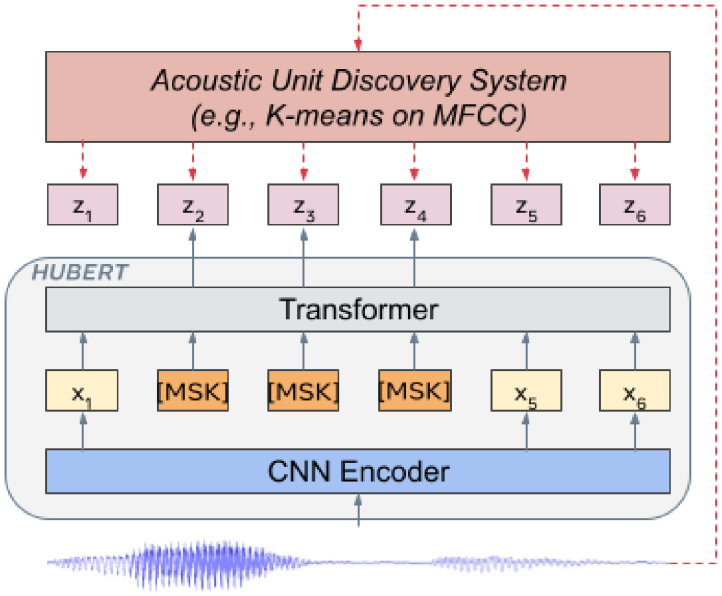


**Figure 3.6.** The HuBERT model predicts hidden cluster assignments of the masked frames generated by k-means clustering using the mel frequency cepstrum coefficient [19].

HuBERT model either matches or improves upon the state-of-the-art wav2vec 2.0 performance [19].

## 3.4 Text emotion recognition

In recent years, text emotion detection as well as speech emotion detection become very popular due to its potential applications in artificial intelligence, human-computer interaction, psychology, marketing, and many others [20].

As well as for speech emotion recognition there was firstly used classification methods like SVM and KNN, also popular method was the Naive Bayes classifier. After that neural networks came.

Because the recognition of emotion from text is based on the words and their context, there can occur following problems and challenges [20]. These problems make emotion recognition from text less accurate in many cases in comparison with speech emotion recognition, which is usually not based on the content.

- **Inability to recognize sentiment.** Some words may cause ambiguity because they have different meanings in different contexts. Also, we can use different types of negations, which can reverse meaning. Both these situations can bring confusion to the recognition model.
- **Identifying different emotions from non-standard language.** In every language there are commonly used slang or informal words, shortcuts, hashtags, and also misspelled words that can often appear in real text conversations. All these can be challenging for the recognition model.
- **Harder prediction of intensity level.** Another problem is that it can be often hard to recognize the intensity of the emotion from text only. Of course, the written text has words that can express intensity, but they are limited.

Anyway, the biggest problems are sentences, where no single word expresses emotion. We have already illustrated this issue at the beginning of this work with the example of the following sentence: `There is a dog.` Only by seeing this text, we do not have any chance to recognize what emotion it contains and probably by text model it would be classified as neutral. But in reality, from the voice, we could be able to recognize for example happiness, because the author really likes dogs, or fear because the author is scared of dogs, etc.

# Chapter 4
# Datasets

First, we will introduce English and Czech datasets usable primarily for speech emotion recognition tasks. There are many datasets publicly available, we have chosen the five most used English datasets and two Czech datasets for comparing state-of-the-art models in the next chapter. These in total seven datasets differ in many important parameters, which will make our tests more varied. Below you can see listed in alphabetical order firstly all five English datasets followed by two Czech datasets. All of them we have analyzed and described with their parameters.

For each dataset, we have plotted how many samples for each class the dataset contains, a histogram of the length of the records, its maximal amplitude, and loudness distribution.

## 4.1 English datasets

### 4.1.1 CREMA-D

CREMA-D dataset [21] contains 12 different sentences. The semantic content of all 12 sentences was rated as emotionally neutral in a prior study. The 12 sentences are:

It's 11 o'clock.

That is exactly what happened.

I'm on my way to the meeting.

I wonder what this is about.

The airplane is almost full.

Maybe tomorrow it will be cold.

I would like a new alarm clock.

I think I have a doctor's appointment.

Don't forget a jacket.

I think I've seen this before.

The surface is slick.

We'll stop in a couple of minutes.

All of these sentences have been vocalized by 91 actors in five different emotions and the neutral state. Several racial and ethnic backgrounds were represented in the actor group: Caucasian, African American, Hispanic, and Asian.

More characteristics are arranged in the table below.

| | |
|---|---|
| dataset full name: | Crowd-sourced Emotional Multimodal Actors Dataset |
| authors: | H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma |
| source: | `https://www.kaggle.com/datasets/ejlok1/cremad` |
| number of records: | 7442 files |
| spoken by: | 91 actors, 48 male and 43 female (aged from 20 to 74 years) |
| type: | single channel of 16000 Hz .wav file |
| average duration: | 2.5 seconds |
| number of speech classes: | 6 |
| classes of speech: | angry, disgusted, fearful, happy, neutral, sad |
| language: | various accent English |
| release date: | december 2014 |

**Table 4.1.** Basic characteristics of CREMA-D dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset.

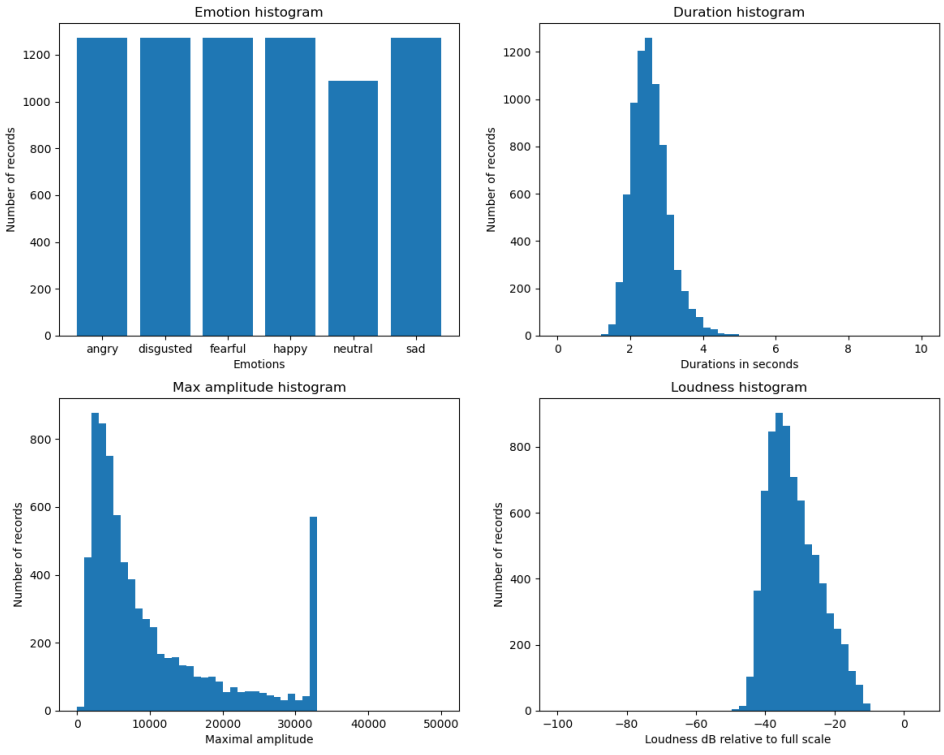


**Figure 4.1.** Dataset characteristics plotted.

### 4.1.2 IEMOCAP

The IEMOCAP dataset [22] is one of the biggest and most important datasets nowadays used for emotion recognition tasks. It contains approximately 12 hours of audiovisual

data, including video, speech, motion capture of face, and text transcriptions, which we will use to test the combination of a text emotion recognition model and a speech emotion recognition model later. It consists of dyadic sessions where actors perform improvisations or scripted scenarios, specifically selected to elicit emotional expressions, what is a big difference in comparison with other datasets, is that in this dataset are not repeated still a few same sentences, but it brings full context dialogues.

It is performed by 10 actors and afterwise each utterance is annotated by at least 3 annotators and one of nine classes was assigned. If no majority ground truth could be assigned, the ground truth was labeled xxx. In our tests, we will not use these items.

More characteristics are arranged in the table below.

| | |
|---|---|
| dataset full name: | The Interactive Emotional Dyadic Motion Capture database |
| authors: | C. Busso, M. Bulut, C.C. Lee, A. Kazemzadeh and others |
| source: | `https://sail.usc.edu/iemocap/index.html` |
| number of records: | 7532 .wav files of one sentence (excluding xxx) |
| spoken by: | 10 Actors: 5 male and 5 female |
| type: | single channel of 16000 Hz .wav file |
| average duration: | 4.6 seconds |
| number of classes: | 9 |
| classes: | anger, happiness, excitement, sadness, frustration, fear, surprise, other, neutral |
| language: | English |
| release date: | December 2008 |

**Table 4.2.** Basic characteristics of IEMOCAP dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset.

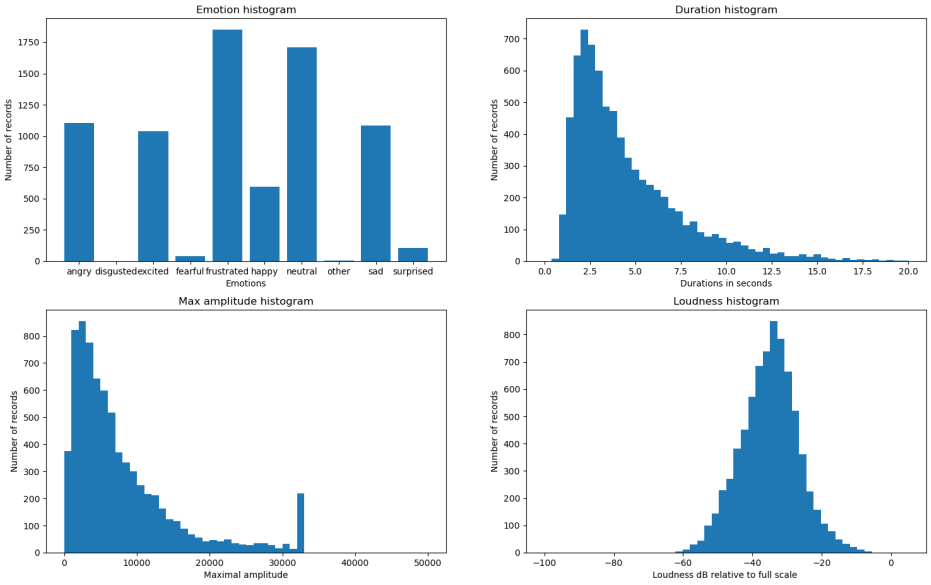


**Figure 4.2.** IEMOCAP dataset characteristics plotted.

### 4.1.3 RAVDESS

Dataset RAVDESS [23] consists only of two lexically-matched sentences:

Kids are talking by the door. Dogs are sitting by the door.

These two sentences are vocalized by 24 professional actors in 7 emotions and the neutral state. This dataset also provides video for each record, but it is not necessary for our purposes. Also, it contains a singing part, but we will not use it either.

More characteristics are arranged in the table below.

| | |
|---|---|
| dataset full name: | The Ryerson Audio-Visual Database of Emotional Speech and Song |
| authors: | Steven R. Livingstone, Frank A. Russo |
| source: | `https://zenodo.org/records/1188976` |
| number of records: | 1440 speech files + 1012 song files |
| spoken by: | 12 male and 12 female professional actors |
| type: | single channel of 48000 Hz .wav file |
| average duration: | 3.7 seconds |
| number of speech classes: | 8 |
| classes of speech: | angry, disgusted, fearful, happy, neutral, sad, surprised, calm |
| number of song classes: | 6 |
| classes of song: | angry, fearful, happy, neutral, sad, calm |
| language: | English (neutral North American accent) |
| release date: | April 2018 |

**Table 4.3.** Basic characteristics of RAVDESS dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset.

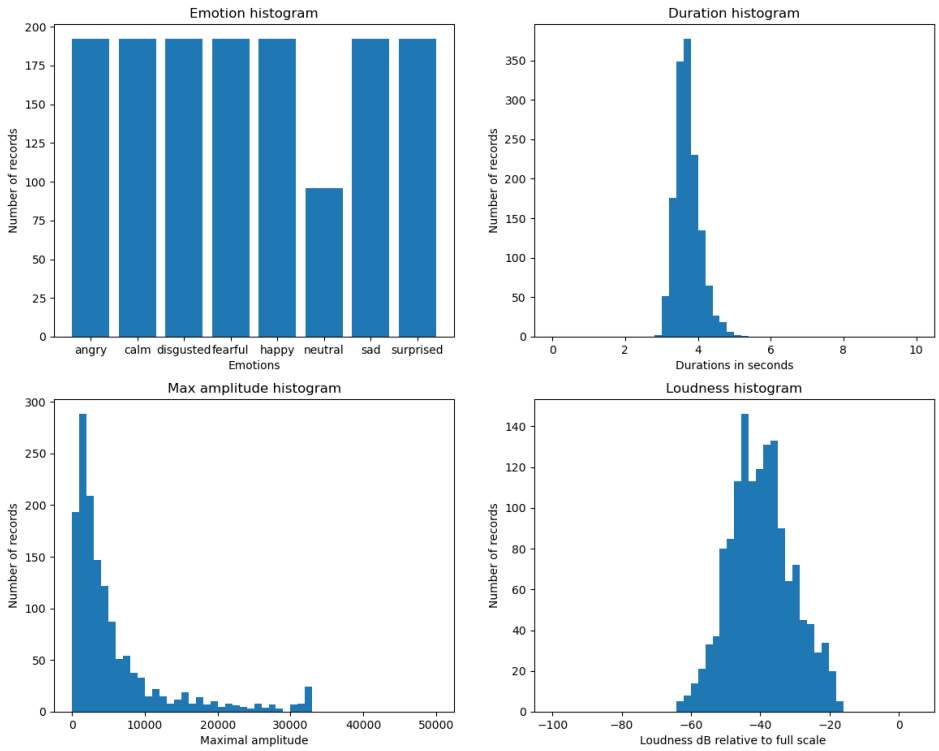


**Figure 4.3.** RAVDESS dataset characteristics plotted.

### 4.1.4 SAVEE

The SAVEE dataset [24] consists of 480 British English utterances in total. There are 8 chosen phonetically-balanced sentences vocalised by 4 male actors. One of them is in 7 different emotions including a neutral state. Others were selected only for one emotion. This dataset contains also visual data, but we will not use them in this work.

These 8 sentences are following:

Common: She had your dark suit in greasy wash water all year. Anger: Who authorized the unlimited expense account?
Disgust: Please take this dirty table cloth to the cleaners for me.
Fear: Call an ambulance for medical assistance.
Happiness: Those musicians harmonize marvelously.
Sadness: The prospect of cutting back spending is an unpleasant one for any governor.
Surprise: The carpet cleaners shampooed our oriental rug.
Neutral: The best way to learn is to solve extra problems.

More characteristics are arranged in the table below.

| | |
|---|---|
| dataset full name: | Surrey Audio-Visual Expressed Emotion |
| authors: | Philip Jackson and Sanaul Haq |
| source: | `http://kahlan.eps.surrey.ac.uk/savee/Database.html` |
| number of records: | 480 files |
| spoken by: | 4 male postgraduate students and researchers (aged from 27 to 31 years) |
| type: | single channel of 44100 Hz .wav file |
| average duration: | 3.8 seconds |
| number of speech classes: | 7 |
| classes of speech: | angry, disgusted, fearful, happy, neutral, sad, surprised |
| language: | British English |
| release date: | 2008 |

**Table 4.4.** Basic characteristics of SAVEE dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset.

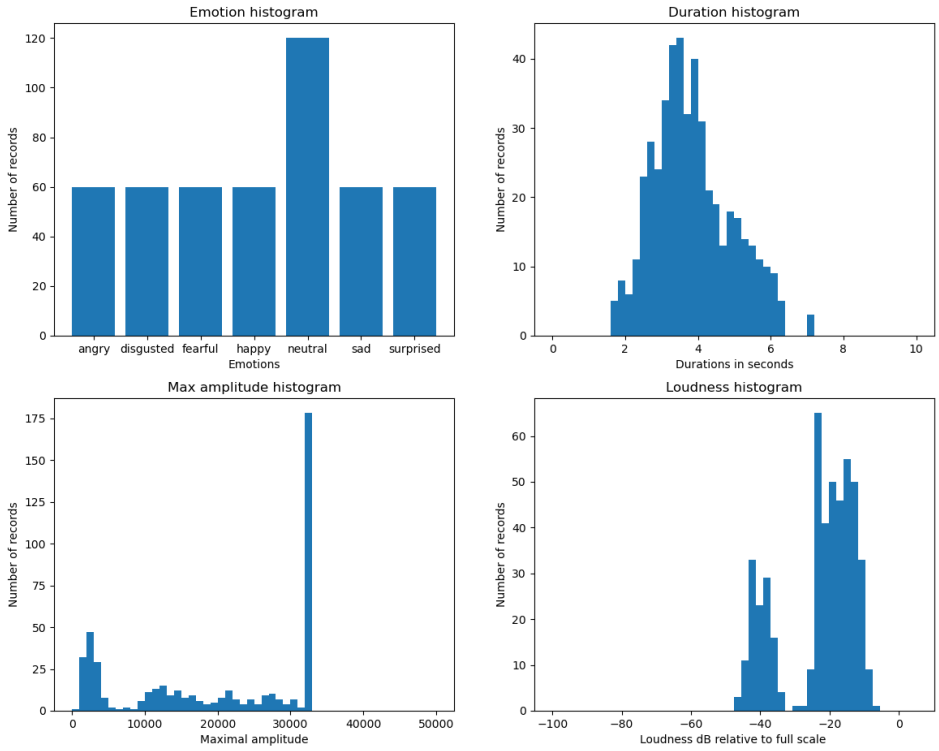


**Figure 4.4.** SAVEE dataset characteristics plotted.

### 4.1.5 TESS

In the TESS dataset [25] there is a set of 200 target words, which were spoken in the carrier phrase: Say the word `[one of that 200 words]`. It was vocalized by two different-aged actresses in 8 different emotions including the neutral state.

More characteristics are arranged in the table below.

| | |
|---|---|
| dataset full name: | Toronto emotional speech set |
| authors: | Pichora-Fuller, M. Kathleen and Dupuis, Kate |
| source: | `https://tspace.library.utoronto.ca/handle/1807/24487` |
| number of records: | 2800 files |
| spoken by: | 2 female actors (aged 26 and 64 years) |
| type: | single channel of 24414 Hz .wav file |
| average duration: | 2.1 seconds |
| number of speech classes: | 7 |
| classes of speech: | angry, disgusted, fearful, happy, neutral, sad, surprised |
| language: | English (Toronto area accent) |
| release date: | 2020 |

**Table 4.5.** Basic characteristics of TESS dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset.

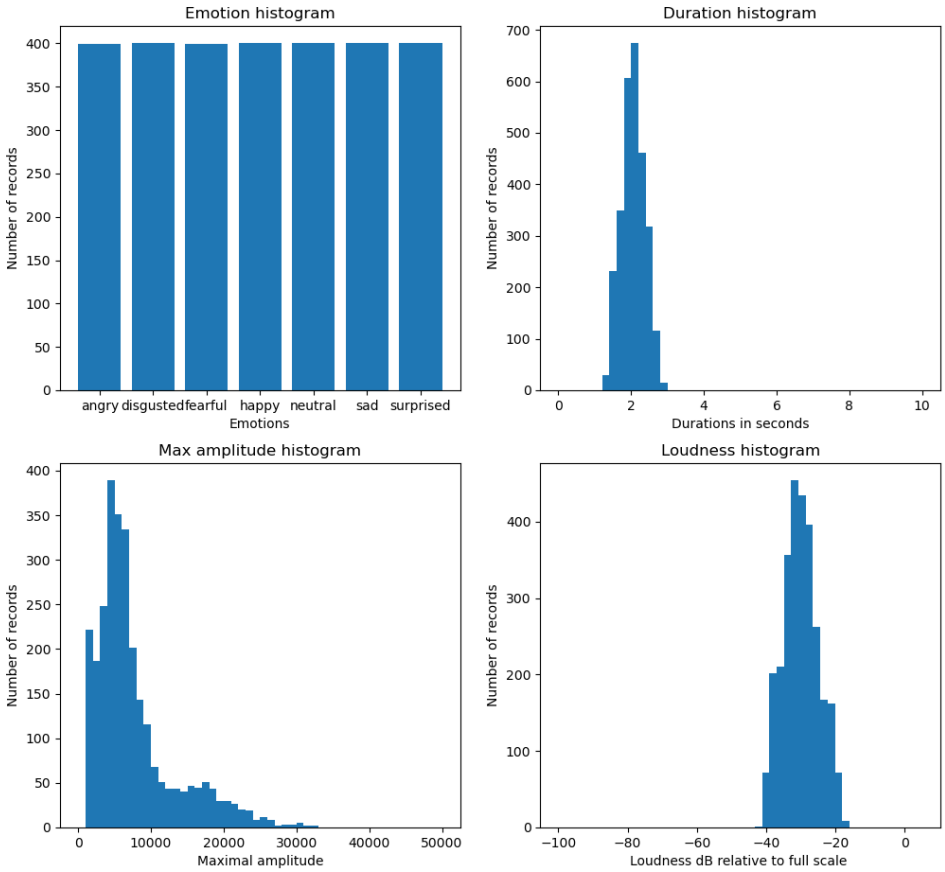


**Figure 4.5.** Tess dataset characteristics plotted.

## 4.2 Czech datasets

### 4.2.1 EmoDBova

This dataset developed by a team led by Dominik Uhrinn at the Technical University of Ostrava [26], contains recordings cut out of television and radio shows.

Some of the recordings have been available on the official archives of radio stations. Some of the television show broadcasting were downloaded from video portals like YouTube. Out of television show broadcasting, they used only sound part for the creation of speech samples [26].

The advantage of this dataset is, that it was subjectively evaluated. Subjective methods mean using people to evaluate samples. The evaluation of database samples was made by students in the age range from 18 to 26 years. Each sample was evaluated by around 16 students, which was supposed to label each sample with one of 5 emotions. These five emotions are angry, fearful, happy, sad, and neutral.

Based on this evaluation, they defined three levels of veracity for each sample: low, medium, and high. The low level has a range from 0 to 80% and represents samples that are not adequately validated or it is harder for a subject to determine what emotional state a sample contains. The medium level range is from 80 – 90% and the high level has a range from 90 to 100%. 100 percent means that all evaluators agreed on one emotion.

The distribution of samples in the dataset according to these categories follows.

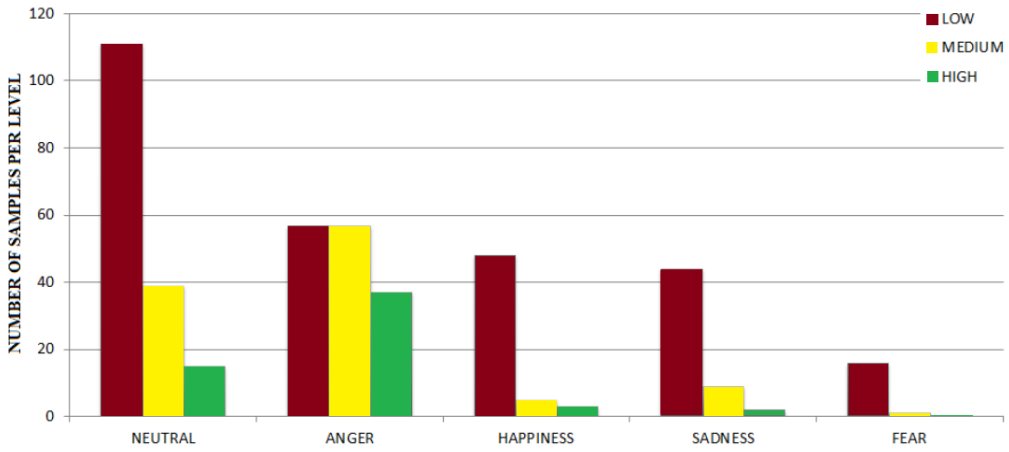


**Figure 4.6.** A number of samples per emotion state and per veracity level [26].

It is important to say, that for our future tests, we have decided to use only samples with veracity high and medium, so with percentages equal and higher than 80%. These two categories include 168 samples.

Characteristic of the whole dataset is described in the following table.

| | |
|---|---|
| dataset full name: | emoDBova |
| authors: | Dominik Uhrin and collective |
| source: | - (not publicly available found) |
| number of records: | 444 files |
| spoken by: | various unknown people |
| type: | 1 channel of 16000 Hz .wav file |
| average duration: | 2.8 seconds |
| number of speech classes: | 5 |
| classes of speech: | angry, fearful, happy, neutral, sad |
| language: | Czech |
| release date: | 2016 |

**Table 4.6.** Basic characteristics of emoDBova dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset, but only of these 168 samples we will work with.

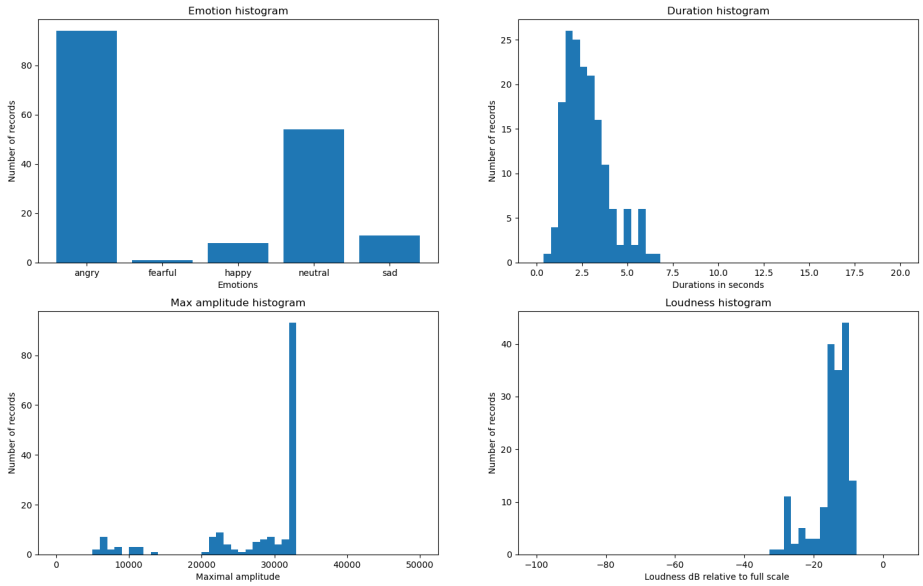


**Figure 4.7.** EmoDBova dataset characteristics plotted. Included are only samples we have decided to work with.

### 4.2.2 EmoMovieDB

This dataset was created by students within the scope of Multimedia Technologies at the Department of Telecommunications, Technical University of Ostrava [27]. A source of the database is the voice of actors from Czech language movies.

More characteristics are arranged in the table below.

| | |
|---|---|
| dataset full name: | emoMovieDB |
| authors: | Pavol Partila and collective |
| source: | `https://dspace.vsb.cz/handle/10084/116855` |
| number of records: | 680 files |
| spoken by: | various Czech actors |
| type: | 2 channel of 16000 Hz .wav file |
| average duration: | 4 seconds |
| number of speech classes: | 5 |
| classes of speech: | angry, fearful, happy, neutral, sad |
| language: | Czech |
| release date: | 2016 |

**Table 4.7.** Basic characteristics of emoMovieDB dataset.

In the following charts, you can see a deeper analyzed distribution of recordings contained in the dataset.

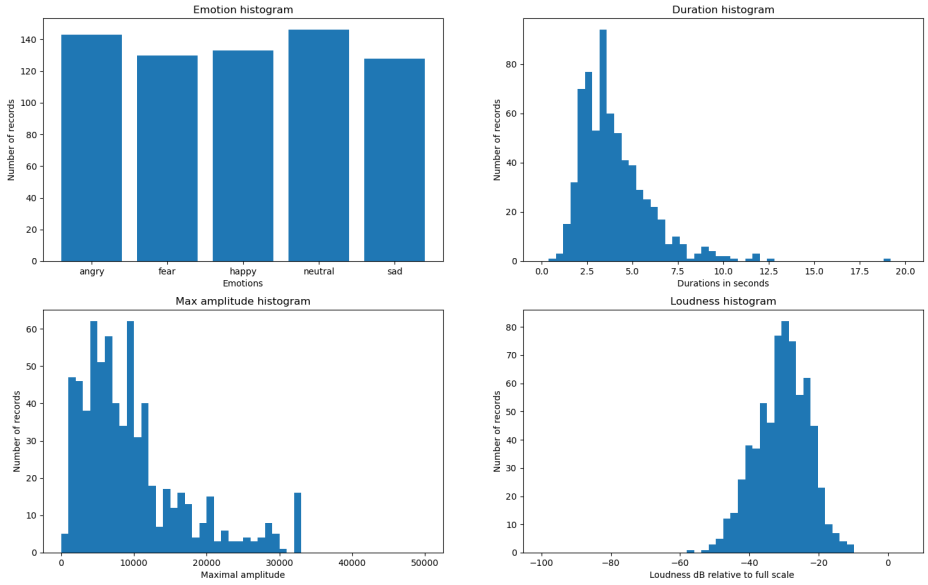


**Figure 4.8.** EmoMovieDB dataset characteristics plotted.

## 4.3 Datasets loading

All of these datasets listed above contain audio signals saved in a .wav file, but the annotation system differs a lot across all these datasets. Some datasets use annotation in the name of the recording, another has a special file with its own structure, where it is necessary to search for the label. Also, the schemes of directories differ. In some datasets, you can find all recordings in one folder, in another they are saved separately.

For these reasons, it was necessary to write an algorithm to read each dataset and assign the correct label to each recording. To be able to work with different models and datasets in which are often the same emotions called or labeled differently, we also had to write dictionaries to unit emotion names across the tests.

For this purpose, we have written the Python class `ReadDataset`, which contains functions for loading each dataset. The only parameter of this class constructor is the path where datasets are stored as they had been downloaded.

Methods in this class are:

```
load_CREMA-D()
load_IEMOCAP(transcriptions=False)
load_RAVDESS()
load_SAVEE()
load_TESS()
load_emoDBova(veracity=80):
load_emoMovieDB()
```

Each of these methods has no parameter and returns two Numpy arrays `paths` and `labels` except for two methods. The first of them is `load_IEMOCAP(transcriptions)`, which has parameter transcriptions by default set to False. If it is set to True this function returns three Numpy arrays `paths, labels, transcriptions`.

The second one is `load_emoDBova(veracity)`, which has parameter veracity. By default is this parameter set to 80. From the dataset will be chosen only samples with equal or higher veracity, than this parameter sets.

Below you can see a part of the class code with a simple usage example.

```
...
class ReadDataset:
...
     def load_CREMA_D(self):
        edict = {
            "ANG": "angry",
            "HAP": "happy",
            "SAD": "sad",
            "FEA": "fearful",
            "DIS": "disgusted",
            "NEU": "neutral",
        }

        for dirname, _, filenames in os.walk(
            self.dataset_path+'/CREMA-D'):
            for filename in filenames:
                self.paths = np.append(self.paths,
                    (os.path.join(dirname, filename)).replace("\\", "/"))
                label = filename.split('_')[2]
                self.labels = np.append(self.labels, edict[label])
        print('CREMA-D loaded, total elements: ', len(self.labels))
        return self.paths, self.labels

if __name__ == '__main__':
    #USAGE EXAMPLE
    dataset_path = "../datasets"
    dataset_reader = ReadDataset(dataset_path=dataset_path)
    paths, labels = dataset_reader.load_CREMA_D()
    # labels[i] is correct label for file with paths[i]
```

# Chapter 5
# Models and testing

On the internet, you can find mentions of many different models for speech emotion recognition tasks published in various platforms like GitHub, HuggingFace, or different storages. Unfortunately, most of them are poorly described and some do not even have code publicly available. Usually, there is also testing on various datasets missing.

This work is mostly focused on emotion recognition from speech, so we will start with models for speech emotion recognition, but then we will slightly introduce also one text recognition model.

Because emotions provided by datasets are not consistent with emotions recognized by selected models, in every test we will input only samples labeled by emotion that the model is supposed to recognize.

## 5.1 Speech emotion recognition models

The goal of this work is to search through all these mentioned platforms and research articles and find well-working usable models. Then describe them, run them, and test them on various selected datasets from the previous chapter.

First, we will start with setting some baseline.

### 5.1.1 My wav2vec

In this case, we will use a pre-trained model [28] from February 2022 developed for dimensional speech emotion recognition based on Wav2vec 2.0. So the original model output is arousal, dominance, and valence. We will use the output from hidden states of this model to get a compact representation of features of input voice recording including information about contained emotions. This procedure is inspired by a Jupyter notebook [29] attached to this model. So from each recording, we will get a 1024-dimensional feature vector which we can classify and assign one of the emotions to it.

But first, we need a classifier. In this case, we will use the support vector classifier provided in the scikit-learn library for Python. First, we have to train this classifier. As a training dataset, we have picked CREMA-D because it provides various sentences spoken by 91 different actors, so we can say it is enough varied.

We have also experimented with training classifiers with raw dataset files and with normalized files in order to improve results.

Trained classifiers we have saved for future use into the files `support_vectors.joblib` and `norm_support_vectors.joblib`.

Implementation of the described algorithm can be found in the file `my_w2v2.py`. This file contains methods: `train_classificator()`, and `predict()` and shows how to get features from an input file.

Now the model and classifier are ready, so we can test it, results can be seen below:

| dataset / model: | trained with normalised data | trained with raw data |
|---|---|---|
| CREMA-D T | 86.51% | 86.51% |
| IEMOCAP | 23.72 | 23.70% |
| RAVDESS | 39.30% | 39.30% |
| SAVEE | 35.48% | 35.24% |
| TESS | 36.07% | 36.03% |
| emoDBova | 13.10% | 13.10% |
| emoMovieDB | 30.44% | 30.29% |

**Table 5.1.** This table presents all test results, representing how many samples were classified correctly in percentage, performed with our finetuned wav2vec model on previously selected datasets. The first 5 rows are English datasets and under the line follows the results on two Czech datasets. Symbol T after a dataset name says that this dataset was used for training of this model. In the left column are dataset names, in the middle one are located results for tests in which input data were normalized and the classifier trained on normalized data was used, and in the last column are results for tests in which input data was not additionally modified and classifier trained on non modified data was used. Each dataset was controlled if the samples had the correct sample rate and number of channels, required by the model, if not they were resampled, and the channel number was set to 1. For the tests, were used only the samples labeled by the emotions for which the model was trained.

As you can see in this case normalization of training data did not bring a big difference. For better result analysis we can take a look at the confusion matrix. We are presenting confusion matrices for the best score and the worst score.

For English datasets:

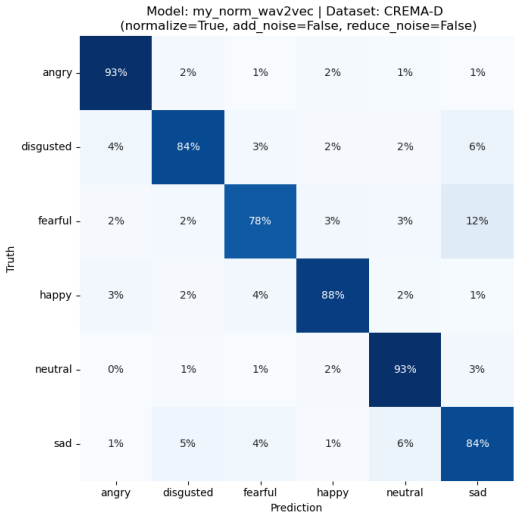


**Figure 5.1.** The confusion matrix in this Figure shows the result of the model My wav2vec trained with normalized dataset tested on CREMA-D dataset. In total, in this test were correctly classified 86.51% samples, which is the best score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.

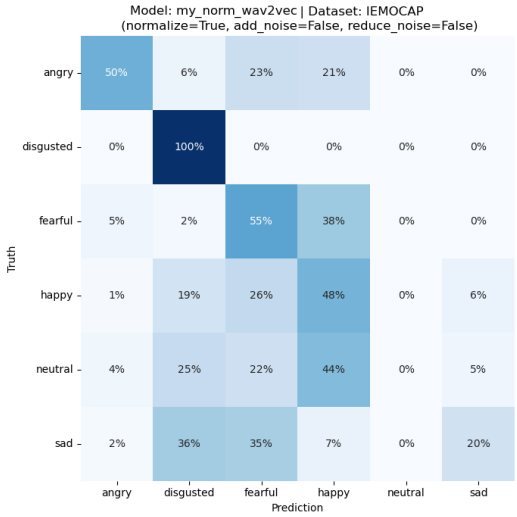


**Figure 5.2.** The confusion matrix in this Figure shows the result of the model My wav2vec trained with normalized dataset tested on IEMOCAP dataset. In total, in this test were correctly classified 23.72% samples, which is the worst score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.

For Czech datasets:

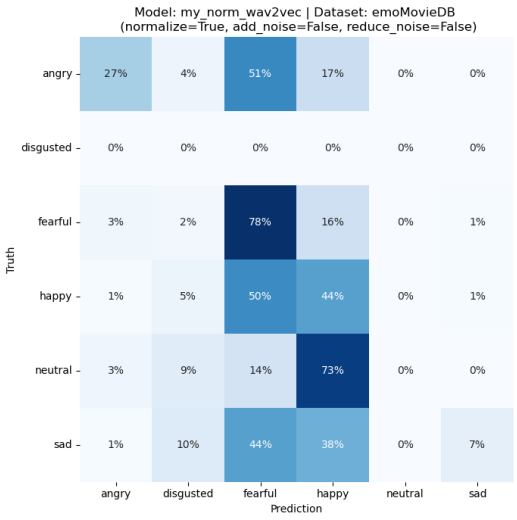


**Figure 5.3.** The confusion matrix in this Figure shows the result of the model My wav2vec trained with normalized dataset tested on emoMovieDB dataset. In total, in this test were correctly classified 30.44% samples, which is the best score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros. Also, you can see columns full of zeros, which is caused by rounding of results.

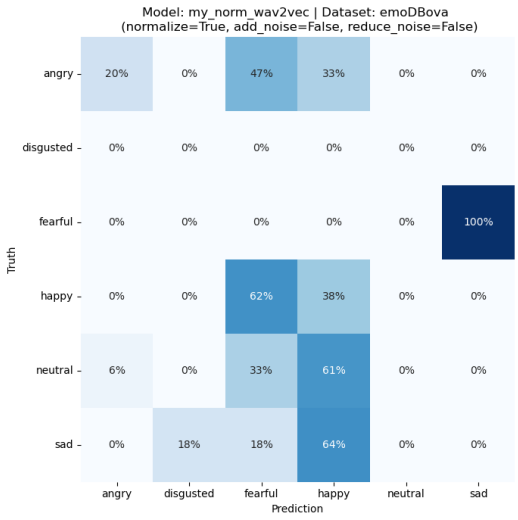


**Figure 5.4.** The confusion matrix in this Figure shows the result of the model My wav2vec trained with normalized dataset tested on emoDBova dataset. In total, in this test were correctly classified 13.10% samples, which is the worst score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros. Also, you can see columns full of zeros, which is caused by rounding of results.

## 5.2 Ehcalabres wav2vec2

This Ehcalabres wav2vec2 model developed by Enrique Hernández Calabrés was published on Hugging Face in 2024 [30]. This model is a fine-tuned version of a model developed for speech recognition [31], which was based on model Wav2Vec2-XLSR-53 developed by Meta.

For fine-tuning of Ehcalabres wav2vec2 was used RAVDESS dataset so it can distinguish 8 emotion classes.

More characteristics are arranged in the following table.

| | |
|---|---|
| model full name: | wav2vec2-lg-xlsr-en-speech-emotion-recognition |
| source: | `huggingface.co/ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition` |
| input type: | single channel of 16000 Hz .wav file |
| number of output classes: | 8 |
| output classes: | angry, calm, disgust, fearful, happy, neutral, sad, surprised |
| training datasets: | RAVDESS |
| training language: | English |
| release date: | 2024 |

**Table 5.2.** Basic characteristics of Ehcalabres wav2vec2 model.

With the model, we have done tests on all previously selected datasets. We have tried if the model will perform better if we first normalize the input data or not. The results of all these tests are in the following Table.

| dataset / model: | tested on normalised data | tested on raw data |
|---|---|---|
| CREMA-D | 38.87% | 38.85% |
| IEMOCAP | 27.23% | 27.18% |
| RAVDESS `T` | 91,46% | 91.81% |
| SAVEE | 38.96% | 38.75% |
| TESS | 37.35% | 37.35% |
| emoDBova | 8.93% | 8.93% |
| emoMovieDB | 30.74% | 30.74% |

**Table 5.3.** This table presents all test results, representing how many samples were classified correctly in percentage, performed with the ehcalabres wav2vec2 model on previously selected datasets. The first 5 rows are English datasets and under the line follows the results on two Czech datasets. Symbol `T` after a dataset name says that this dataset was used for training of this model. In the left column are dataset names, in the middle one are located results for tests in which input data were normalized, and in the last column are results for tests in which input data was not additionally modified. Each dataset was controlled if the samples had the correct sample rate and number of channels, required by the model, if not they were resampled, and the channel number was set to 1. For the tests, were used only the samples labeled by the emotions for which the model was trained.

For a deeper understanding of model behavior and results presented in the previous Table, now we will show two confusion matrices in which we will be able to see how different emotions were mostly classified. We will show confusion matrices for the best and the worst results according to the previous Table.

For English datasets:

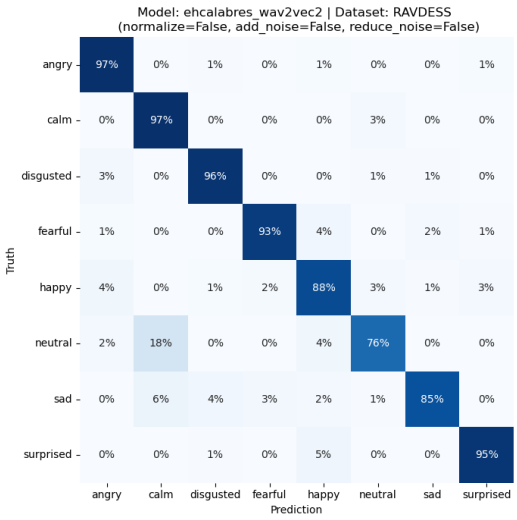


**Figure 5.5.** The confusion matrix in this Figure shows the result of the model ehcalabres wav2vec2 tested on RAVDESS dataset. In total, in this test were correctly classified 91.81% samples, which is the best score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.

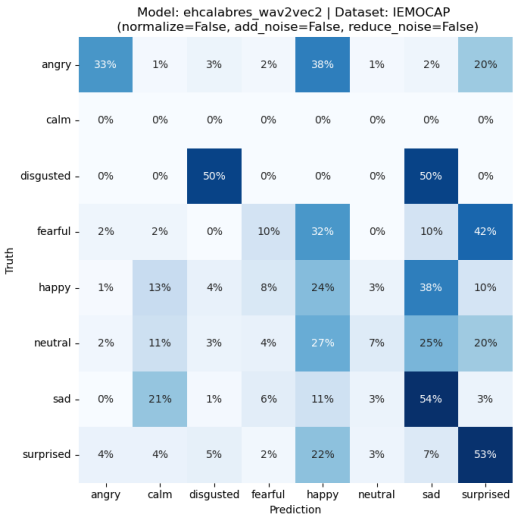


**Figure 5.6.** The confusion matrix in this Figure shows the result of the model ehcalabres wav2vec2 tested on IEMOCAP dataset. In total, in this test were correctly classified 27.18% samples, which is the worst score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros.

For Czech datasets:

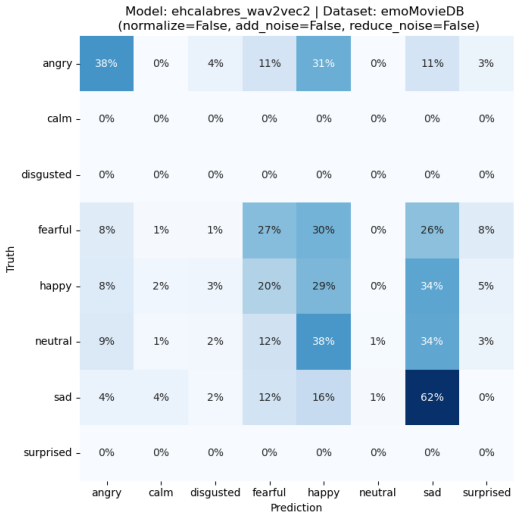


**Figure 5.7.** The confusion matrix in this Figure shows the result of the model ehcalabres wav2vec2 tested on emoMovieDB dataset. In total, in this test were correctly classified 30.74% samples, which is the best score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros. Also, you can see columns full of zeros, which is caused by rounding of results.

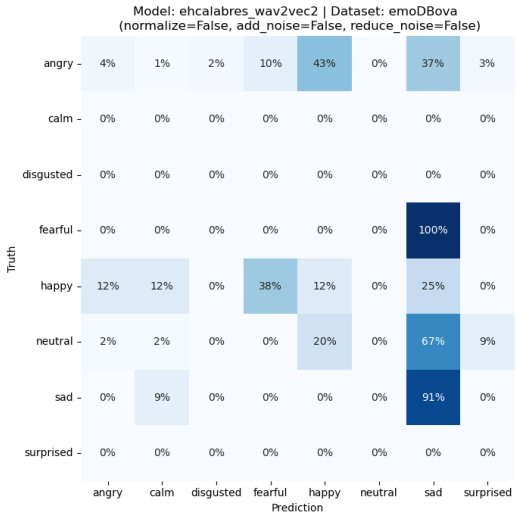


**Figure 5.8.** The confusion matrix in this Figure shows the result of the model ehcalabres wav2vec2 tested on emoDBova dataset. In total, in this test were correctly classified 8.93% samples, which is the worst score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros. Also, you can see columns full of zeros, which is caused by rounding of results.

### 5.2.1 Emotion2vec

This state-of-the-art model published in December 2023 was from the beginning primarily developed for speech emotion recognition [32].

The emotion2vec model was pre-trained on 262 hours of unlabeled emotion data through self-supervised online distillation, leading to universal emotion representation ability [32].

Emotion2vec was also tested on 10 languages, and the results show that this model exhibits a good ability for language generalization [32].

The following diagram introduces the model's training phase and the final part for classification.

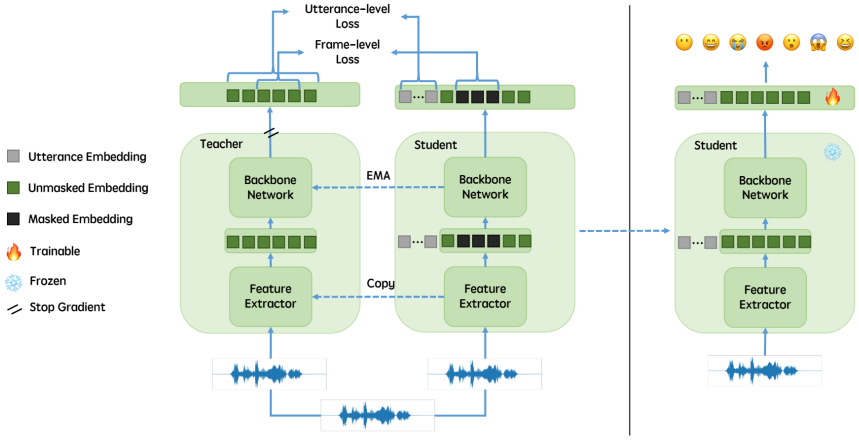


**Figure 5.9.** The overall framework of emotion2vec. During the pre-training phase, emotion2vec conducts online distillation with a teacher network and a student network. When a specific downstream task is performed, emotion2vec is frozen and a lightweight downstream model is trained [32].

In January 2024, the 9-class emotion recognition model fine-tuned from emotion2vec has been released. This fine-tuned model we will use for our testing. Its characteristics can be seen in the following Table.

| | |
|---|---|
| model full name: | emotion2vec_base_finetuned |
| source: | `github.com/ddlBoJack/emotion2vec` |
| input type: | single channel of 16000 Hz .wav file |
| number of output classes: | 9 |
| output classes: | angry, disgusted, fearful, happy, neutral, other, sad, surprised, unknown |
| training datasets: | IEMOCAP, MELD, CMU-MOSEI, MSP-Podcast, MEAD |
| training language: | English |
| release date: | December 2023 |

**Table 5.4.** Basic characteristics of emotion2vec model.

With the model, we have done tests on all previously selected datasets. We have tried if the model will perform better if we first normalize the input data or not. The results of all these tests are in the following Table.

| dataset / model: | tested on normalised data | tested on raw data |
|---|---|---|
| CREMA-D | 80.89% | 80.88% |
| IEMOCAP T | 64.39% | 64.43% |
| RAVDESS | 87.10% | 87.82% |
| SAVEE | 93.33% | 93.54% |
| TESS | 48.07% | 48.14% |
| emoDBova | 41.07% | 41.07% |
| emoMovieDB | 38.82% | 38.97% |

**Table 5.5.** This table presents all test results, representing how many samples were classified correctly in percentage, performed with the emotion2vec model on previously selected datasets. The first 5 rows are English datasets and under the line follows the results on two Czech datasets. Symbol T after a dataset name says that this dataset was used for training of this model. In the left column are dataset names, in the middle one are located results for tests in which input data were normalized, and in the last column are results for tests in which input data was not additionally modified. Each dataset was controlled if the samples had the correct sample rate and number of channels, required by the model, if not they were resampled, and the channel number was set to 1. For the tests, were used only the samples labeled by the emotions for which the model was trained.

For a better understanding of model behavior and results presented in the previous Table, now we will show two confusion matrices in which we will be able to see how different emotions were mostly classified. We will show confusion matrices for the best and the worst results according to the previous Table.

For English datasets:

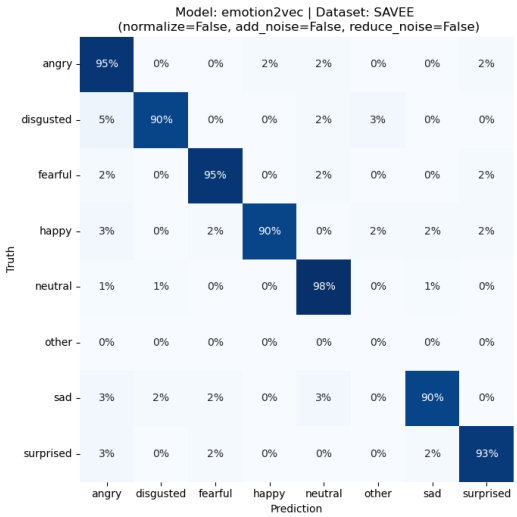


**Figure 5.10.** The confusion matrix in this Figure shows the result of the model emotion2vec tested on SAVEE dataset. In total, in this test were correctly classified 93.54% samples, which is the best score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros.

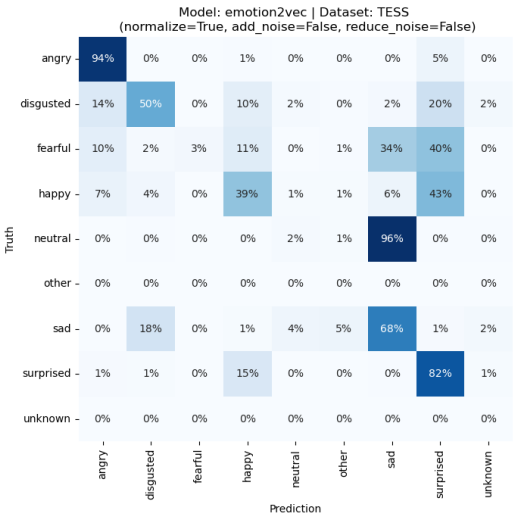


**Figure 5.11.** The confusion matrix in this Figure shows the result of the model emotion2vec tested on TESS dataset. In total, in this test were correctly classified 48.07% samples, which is the worst score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros.

For Czech datasets:

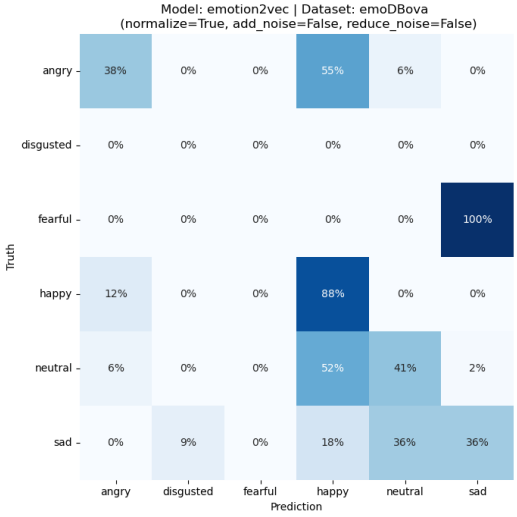


**Figure 5.12.** The confusion matrix in this Figure shows the result of the model emotion2vec tested on emoDBova dataset. In total, in this test were correctly classified 41.07% samples, which is the best score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros.

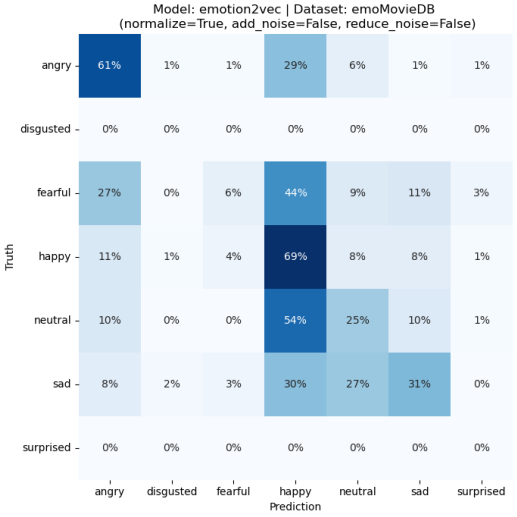


**Figure 5.13.** The confusion matrix in this Figure shows the result of the model emotion2vec tested on emoMovieDB dataset. In total, in this test were correctly classified 38.82% samples, which is the worst score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained. Because in this case, the model distinguishes more emotion classes, than are in datasets, there are rows with only zeros.

### 5.2.2 S3prl hubert

This model published also on Hugging Face is a model based on model hubert-large-ll60k developed and pretreated by Meta [33]. The IEMOCAP dataset was used for fine-tuning of this model. Basic characteristics can be seen in the following Table.

| | |
|---|---|
| model full name: | hubert-large-superb-er |
| source: | `huggingface.co/superb/hubert-large-superb-er` |
| input type: | single channel of 16000 Hz .wav file |
| number of output classes: | 4 |
| output classes: | angry, happy, neutral, sad |
| training datasets: | IEMOCAP |
| training language: | English |
| release date: | 2021 |

**Table 5.6.** Basic characteristics of s3prl hubert model.

With the model, we have done tests on all previously selected datasets. We have tried if the model will perform better if we first normalize the input data or not. The results of all these tests are in the following Table.

| dataset / model: | tested on normalised data | tested on raw data |
|---|---|---|
| CREMA-D | 45.27% | 54.78% |
| IEMOCAP `T` | 61.47% | 67.48% |
| RAVDESS | 29.61% | 32.44% |
| SAVEE | 40.00% | 45.00% |
| TESS | 53.16% | 59.04% |
| emoDBova | 20.96% | 20.96% |
| emoMovieDB | 35.27% | 36.18% |

**Table 5.7.** This table presents all test results, representing how many samples were classified correctly in percentage, performed with the s3prl hubert model on previously selected datasets. The first 5 rows are English datasets and under the line follows the results on two Czech datasets. Symbol `T` after a dataset name says that this dataset was used for training of this model. In the left column are dataset names, in the middle one are located results for tests in which input data were normalized, and in the last column are results for tests in which input data was not additionally modified. Each dataset was controlled if the samples had the correct sample rate and number of channels, required by the model, if not they were resampled, and the channel number was set to 1. For the tests, were used only the samples labeled by the emotions for which the model was trained.

For a better understanding of model behavior and results presented in the previous Table, now we will show two confusion matrices in which we will be able to see how different emotions were mostly classified. We will show confusion matrices for the best and the worst results according to the previous Table.

For English datasets:

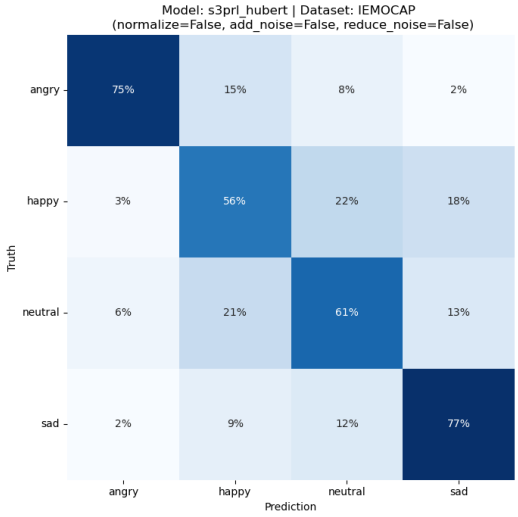


**Figure 5.14.** The confusion matrix in this Figure shows the result of the model s3prl hubert tested on IEMOCAP dataset. In total, in this test were correctly classified 67.48% samples, which is the best score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.

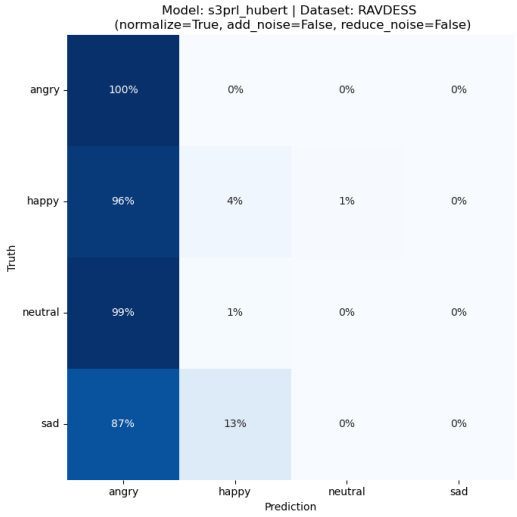


**Figure 5.15.** The confusion matrix in this Figure shows the result of the model s3prl hubert tested on RAVDESS dataset. In total, in this test were correctly classified 29.61% samples, which is the worst score for this model across all English datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.

For Czech datasets:

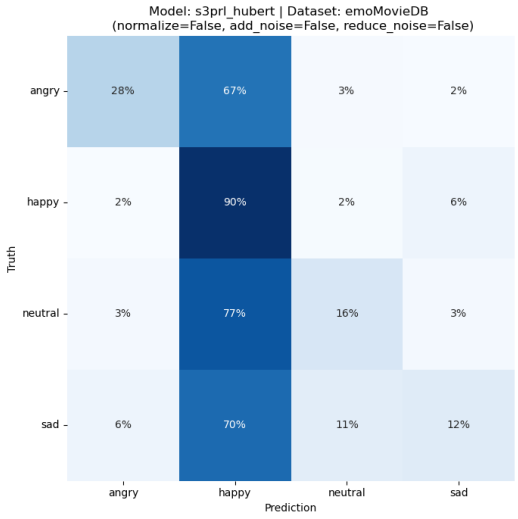


**Figure 5.16.** The confusion matrix in this Figure shows the result of the model s3prl hubert tested on emoMovieDB dataset. In total, in this test were correctly classified 36.18% samples, which is the best score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.


**Figure 5.17.** The confusion matrix in this Figure shows the result of the model s3prl hubert tested on emoDBova dataset. In total, in this test were correctly classified 20.96% samples, which is the worst score for this model across all Czech datasets. It is important to mention, that the confusion matrix is constructed to be a square matrix, where correct results are located on the diagonal. Results are presented in percentages, so the sum of each row should be 100%, but because results are rounded it can cause a different sum. In the matrix are only labels containing at least one prediction. For the test, were used only the samples labeled by the emotions for which the model was trained.

## 5.3 Text emotion recognition model

In this section, we will briefly introduce a model for emotion recognition from text. This model provides the best results according to testing provided by Klára Losenická in her bachelor thesis [34]. We will use this model in the final process chain later.

### 5.3.1 Rahulmallah

The basic characteristics of this model are arranged in the following table.

| | |
|---|---|
| model full name: | rahulmallah/autotrain-emotion-detection-1366352626 |
| source: | `huggingface.co/rahulmallah/autotrain-emotion-detection-1366352626` |
| input type: | string (sentence) |
| number of output classes: | 13 |
| output classes: | anger, boredom, empty, enthusiasm, fun, happiness, hate, love, neutral, relief, sadness, surprise, worry |
| training datasets: | rahulmallah/autotrain-data-emotion-detection |
| training language: | English |
| release date: | 2022 |

**Table 5.8.** Basic characteristics of Rahulmallah model.

## 5.4 Overall results

Overall scores of tests performed in this chapter are presented in the following table. Symbol `T` before score value marks, that this dataset was used for this model training. Value says how many of the samples from the dataset were recognized correctly in percentage. From each dataset for each model, only samples labeled with emotions that the model had been trained to recognize have been selected.

Also, we have taken a look if models have performed better when input was normalized or not. In the previous results, we can see, that the normalization of input data usually had no significant effect. For model s3prl hubert, the results got significantly worse.

In the following table, we show only these better results. If the model has achieved better results with normalized input data, there is a symbol `N` next to the model name in the following table.

| model -> dataset: | My norm wav2vec | ehcalabres wav2vec2 `N` | emotion2vec | s3prl hubert |
|---|---|---|---|---|
| CREMA-D | `T` 86,51% | 38.87% | 80.88% | 54.78% |
| IEMOCAP | 23.72% | 27.23% | `T` 64.43% | `T` 67.48% |
| RAVDESS | 39,30% | `T` 91,46% | 87.82% | 32.44% |
| SAVEE | 35,48% | 38.96% | 93.54% | 45.00% |
| TESS | 36,07% | 37.35% | 48.14% | 59.04% |
| emoDBova | 13.10% | 8.93% | 41.07% | 20.96% |
| emoMovieDB | 30.44% | 30.74% | 38.97% | 36.18% |

**Table 5.9.** Models results over all datasets comparison table.

From the table we can clearly see, that emotion2vec model beats all others in both languages. can see that the best speech emotion recognition model is emotion2vec. The second best model is s3prl hubert. Both models based on wav2vec 2.0 model performed poorly in both languages. We can only discuss, that poor results of excalabres wav2vec are caused also by the training dataset - RAVDESS, which is not so complex in comparison with for example IEMOCAP dataset.

For future experiments and the final process chain, we will pick the best model, which is emotion2vec.

# Chapter 6
# Further experiments

## 6.1 Input data processing

To be able to make various experiments we have written the Python class `Process` using libraries pydub, numpy and noisereduce. Input is a wav file location and it provides various output possibilities including exporting edited wav file or pydub format audio.

It provides the possibility of normalizing of audio input, adding white noise, or reducing noise in the input file. Also, it can return only the defined time interval of the input file. This class also by default sets the sample rate to 16000 Hz, and the number of channels to 1, which is input for the most tested models.

The class also contains a few functions for analyzing and visualization of audio using the matplotlib library.

An example of the core part of the class implementation follows.

```
class Process:
    def __init__(self, wav_url, sampling_rate=16000,
    normalize=False, add_noise=False, reduce_noise=False):
        self.wav_url = wav_url
        self.wav_file = AudioSegment.from_file(file=wav_url,
        format="wav")

        if add_noise:
            gain = self.get_loudness() - 10.0
            self.wav_file = self.wav_file.overlay(
                WhiteNoise(sample_rate=self.wav_file.frame_rate)
                    .to_audio_segment(duration=len(self.wav_file))
                    .apply_gain(gain))

        if sampling_rate is not None:
            self.wav_file = self.wav_file.set_frame_rate(sampling_rate)

        if reduce_noise:
            reduced = noisereduce.reduce_noise(
                y=self.wav_file.get_array_of_samples(),
                sr=self.wav_file.frame_rate)
            self.wav_file = AudioSegment(np.int16(reduced).tobytes(),
                frame_rate=self.wav_file.frame_rate,
                sample_width=self.wav_file.sample_width, channels=1)

        if normalize:
            self.wav_file = effects.normalize(self.wav_file)
...
```

## 6.2 Experiments

From the previous section and according to results in the Table 5.9, we can see that our best speech emotion recognition model is emotion2vec. Now we will take this model and perform another experiment on it, to get to know how it behaves and if we can increase its success rate by applying noise reduction on input data.

For all experiments in this chapter, we will use the CREMA-D dataset. We have chosen this dataset because it consists of 12 different sentences spoken by 91 actors from various ethnic backgrounds, so we can say, that with its 7442 samples is complex and varied enough.

### 6.2.1 White noise addition

First, we will observe the behavior of this model after adding white noise to the input audio.

White noise is a random signal, which has equal intensity at different frequencies. To generate it we have used a white noise generator contained in the pydub library we are using to work with audio files. Implementation you can see in the code example above. We set the loudness of the white noise 10 dB lower than has the initial sound to simulate the most realistic situation. In the Figures below you can see an example of the original audio file from the CREMA-D dataset and its spectrogram compared with the same audio file after the white noise had been added. In the spectrogram of recording with white noise, all frequencies all contained.

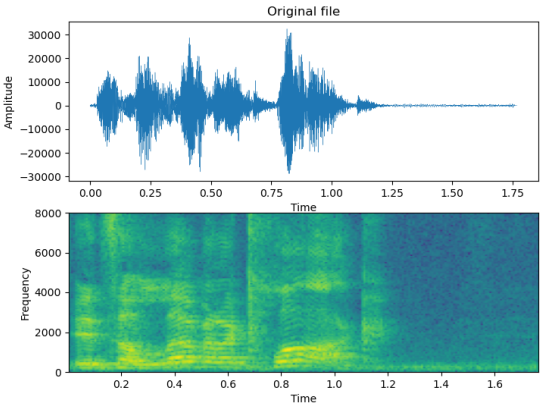


**Figure 6.1.** Before white noise added.

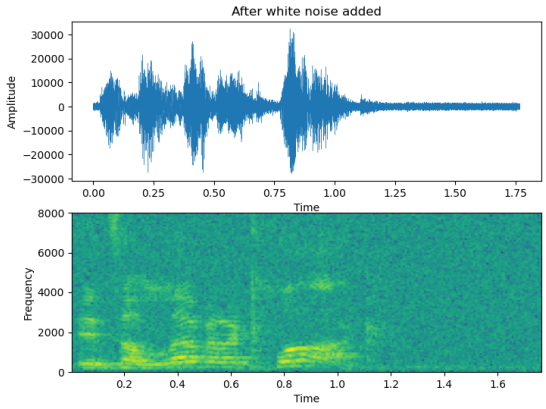


**Figure 6.2.** After white noise added.

We have tested this model with the CREMA-D dataset. Into each file was added white noise. As expected the results got worse. (`Before` says the model result on this dataset before noise modification.) Confusion matrices for both models in this test are the following.

```
Result for emotion2vec:
    Correct: 4605  Incorrect: 2837
    Total success score: 61.88% (before 80.89%)
```

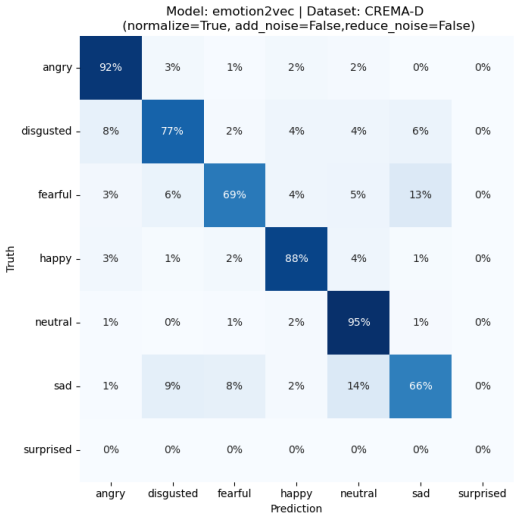


**Figure 6.3.** Emotion2vec before white noise added.

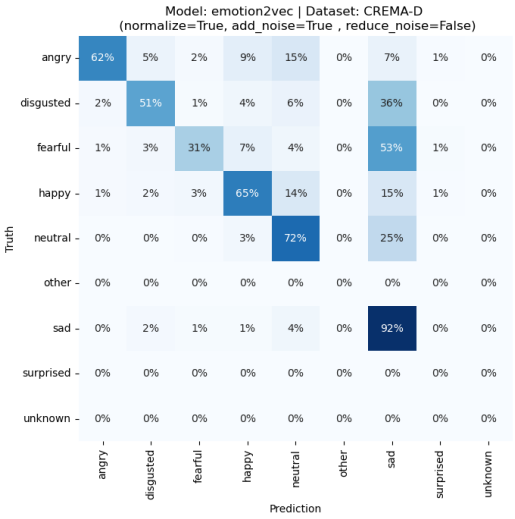


**Figure 6.4.** Emotion2vec after white noise added.

### 6.2.2 Noise reduction

In this test, we will reduce noise in the audio records hoping for better results. To reduce noise we have selected the library noisereduce [35] for Python. There are many libraries and code examples for noise reduction publicly available. We have chosen this one, because it was developed for biological voice signals like speech, bioacoustics, and physiological signals.[36] It is well documented and widely used in speech processing.

The difference between spectrograms and waveforms generated before and after the noise reduction can be seen below.

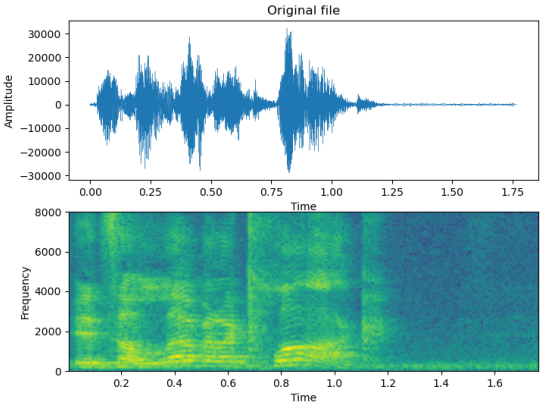


**Figure 6.5.** Before noise reduced.

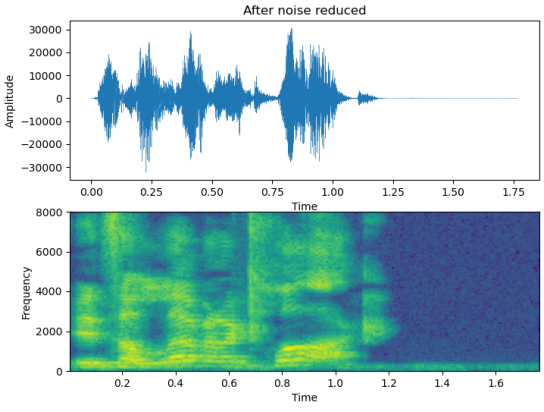


**Figure 6.6.** After noise reduced.

After the noise reduction, the voice record sounds clearer to listen to but as you can see in the spectrogram after noise reduction there are many frequencies disappeared through the record. The influence and importance of these frequencies for the emotion recognition process can be significantly seen in the results below.

```
Result for emotion2vec:
    Correct: 4522  Incorrect: 2920
    Total success score: 60.76% (before 80.89%)
```

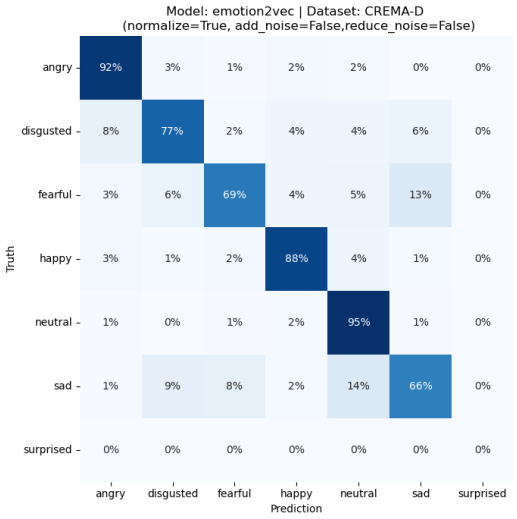


**Figure 6.7.** Emotion2vec before noise reduced.

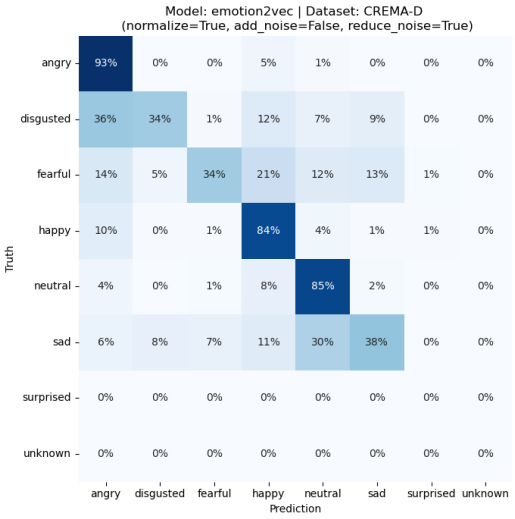


**Figure 6.8.** Emotion2vec after noise reduced.

Although the voice sounds clearer, we can see that the result is worse. The result of the model is even worse then result of the model in the noise addition test. So we can suppose that noise reduction removed also frequencies important for correct emotion recognition, probably model had been trained on the datasets without noise reduction applied.

# Chapter 7
# From text and speech to emotion

The final goal of this work is to describe and implement a way, how to get rational output from a text model, speech model, or combination of both models.

First, we will start with optional data reprocessing, because the input for our process chain, which we will implement later, is supposed to be only short pieces of conversation, first, we need to solve how to make these pieces from longer voice records or longer text.

## 7.1 Speech segmentation and diarization

Apart from basic segmentation to pieces with fixed time or size, we can use some more sophisticated tools, which can segment speech recordings into sentences and even more it can identify different speakers, and assign them to sentences. This process is called speaker diarization.

For this purpose there exist many tools publicly available, we will introduce one of them which is well known and also was tested in our research group and performed well. That one is **Pyannote.audio speaker diarization** [37] [38], which you can find here `https://huggingface.co/pyannote/speaker-diarization`.

Its input is wav file. To try it we will use part of one record from the IEMOCAP dataset, which contains dialog. How the output can look like you can see below:

```
start=6.933s stop=7.338s speaker_SPEAKER_01  len=0.405s
start=9.110s stop=10.072s speaker_SPEAKER_00  len=0.962s
start=10.527s stop=10.865s speaker_SPEAKER_01  len=0.338s
start=11.017s stop=11.033s speaker_SPEAKER_00  len=0.017s
start=11.050s stop=12.147s speaker_SPEAKER_00  len=1.097s
...
```

When listening it contains:
`Excuse me.`
`Do you have your forms?`
`Yeah.`
_ *in next step we can filter too short pieces like this one
`Let me see them.`

This corresponds with dataset transcription.

We will use start and stop times to segment long speech input, optionally we can add padding. Now we know how we can optionally prepare input for our future process chain.

## 7.2 Joining text and speech emotion recognition

### 7.2.1 How to tackle with different emotions

Another challenge is to tackle with different emotions. Because there are so many emotions and approaches to their classification, it is almost impossible to find the speech model and text model recognizing the same set of emotions. Unfortunately for us, there does not exist an emotion map with calculated emotion distances. Therefore we need a tool how to group similar emotions. One of the existing models is **circumplex model** [39]. It is supposed to allow us to divide all emotions along two axes, arousal and valence as you can see in the Figure below. Unfortunately, it does not allow us to divide all emotions into four categories as it might seem at first glance. There are many problematic emotions like a surprise which can be positive or negative and others which is hard to assign. For this reason, we have decided to find another approach.

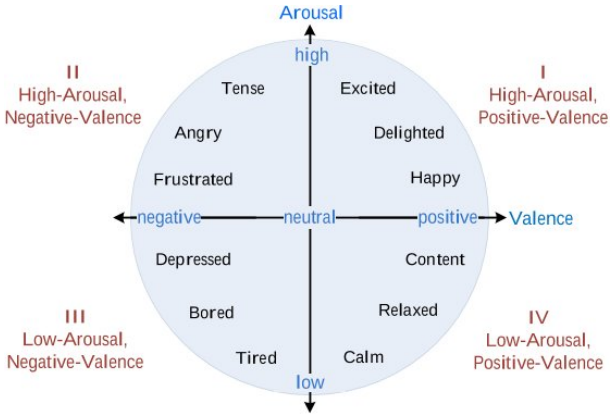


**Figure 7.1.** Circumplex emotion model. [39]

A less problematic division is to divide emotions by **sentiment**. There we have by default three categories: positive, negative, and neutral. We will take inspiration from the sentiment dictionary [40] developed by Google and published on GitHub, which already contains the category ambiguous for emotions which can have sometimes positive and sometimes negative coloration. Then we will add the category unknown and neutral. Also, we have to edit emotion's names to fit our models. The final dictionary you can see below.

```
sentiment_dictionary = {
        "positive": ["happy", "amusement", "excited", "joy", "love",
                     "desire", "optimism", "caring", "pride",
                     "admiration", "gratitude", "relief", "approval",
                     "funny", "enthusiastic", "calm"],
        "negative": ["fearful", "frustrated", "nervousness", "remorse",
                      "embarrassed", "disappointed", "sad", "grief",
                      "disgusted", "angry", "annoyed", "disapproval",
                      "hate", "worried", "bored", "empty"],
        "ambiguous": ["realization", "surprised", "curiosity",
                     "confusion"],
        "neutral": ["neutral"]
        "unknown": ["unknown", "other"]
    }
```

Now we know how to divide emotions and we are ready to test, compare, and finally join our models.

### 7.2.2 Text and speech results fusion

In order to get maximally correct and informative output from the combination of both models, we have created the following decision tree. It takes into account all the information we have already got to know before.

If emotions from both models are the same, that is the best possible option, then the output will be that emotion. If not, then we will take a look if they are both positive or negative, if so we will as an output return both of them to give maximum possible information. On the other hand in the case when one emotion comes from the positive category and the second one from the negative category, then we will output `unknown` to filter the maximum of possible mistakes.

As we already know there are situations, when the text seems neutral, but we can easily identify emotion from the voice. That is why we prioritize speech models when text model output comes from the neutral or ambiguous category. When the opposite situations happen, we can consider oppositely prioritizing the text model's output, but as we can see from tests we have already done, the speech model has better results, so in this situation we will print both results. The described decision tree schematically follows:

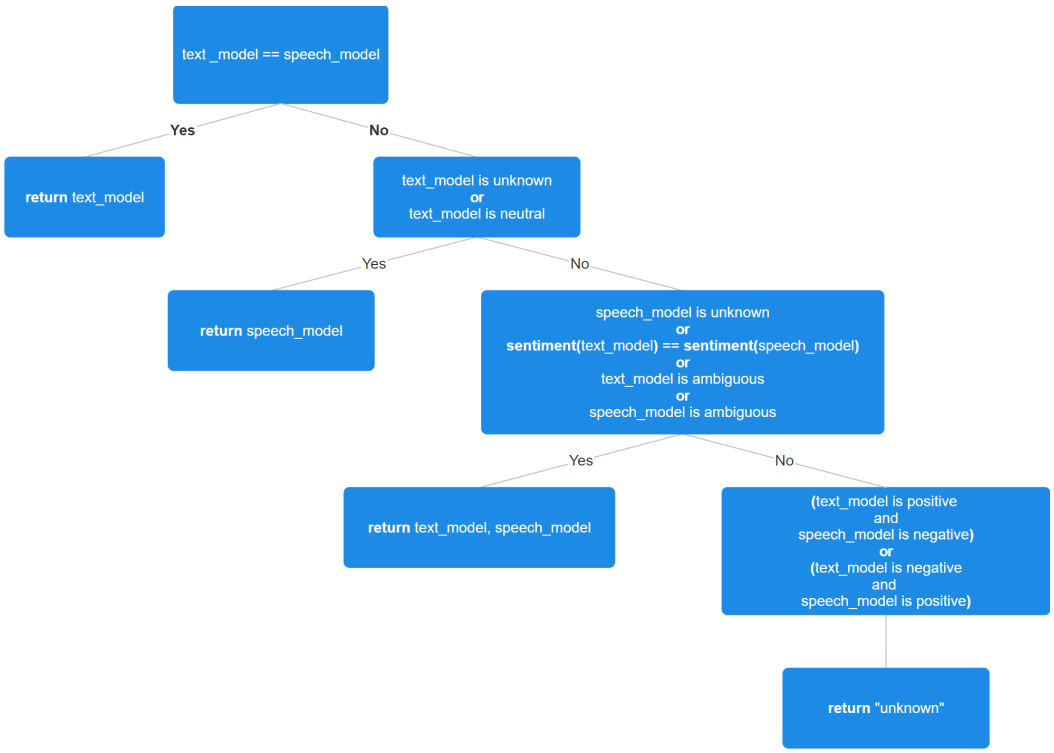


**Figure 7.2.** Decision tree to get optimal output from the results of text and speech models which comes to this tree as two inputs `text_model` and `speech_model`. The notation `sentiment([xxx]_model)` returns one of the five categories in our sentiment dictionary.

After going through this tree we can get one of the following outputs.

- one emotion
- pair of two emotions both from the negative sentiment category

- pair of two emotions both from the positive sentiment category
- pair of two emotions both from ambiguous sentiment category
- pair of emotions from the category positive, negative, or ambiguous and `neutral`
- pair of two emotions one of them from the ambiguous sentiment category and another from the positive or negative category
- `neutral`
- `unknown`

### 7.2.3 Testing and results

For this final testing, we are taking the best speech model and the best text model according to previous testing.

First, we will show results in confusion matrices for both models tested on the IEMOCAP dataset, which contains voice records for the speech model and transcriptions of them for the text model.

Firstly we will use only samples labeled by emotion which the model is trained for.

The results are following:

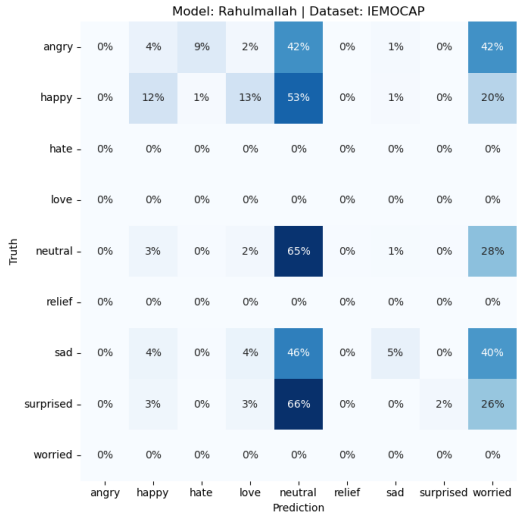


**Figure 7.3.** Confusion matrix showing how text model Rahulmallah classified samples from IEMOCAP dataset. This time there was used only samples labelled by emotions for which this model had been trained for.

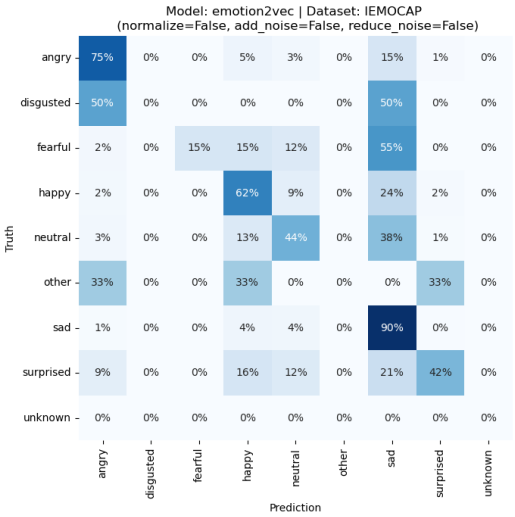


**Figure 7.4.** Confusion matrix showing how speech model Emotion2vec classified samples from IEMOCAP dataset. This time there was used only samples labelled by emotions for which this model had been trained for.

Then we will test how can our models handle all emotions provided in the IEMOCAP dataset, which are anger, happiness, excitement, sadness, frustration, fear, surprise, and neutral so even with emotions for which this model was not trained. We will leave out samples labeled as other.

To remind, the text model was trained to recognize anger, boredom, empty, enthusiasm, fun, happiness, hate, love, relief, sadness, surprise, worry, and neutral.

The speech model was trained to recognize angry, disgusted, fearful, happy, neutral, sad, and surprised. Also speech model can return states unknown or other, which we have decided to unite under unknown.

In the following Figures, you can see confusion matrices. In the left Figure, you can see, how was classified all samples from the IEMOCAP dataset, excluding samples

labeled as other, by text model only. In the right Figure, you can see how the same samples were classified by speech model only.

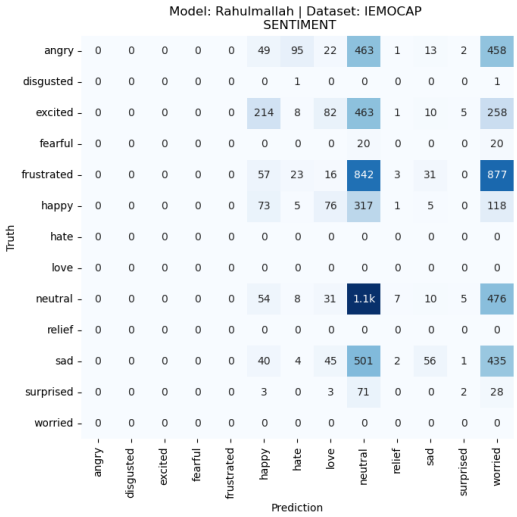


**Figure 7.5.** Confusion matrix showing how text model Rahulmallah classified samples from the IEMOCAP dataset. This time there was used all samples from the IEMOCAP dataset, excluded were samples labeled as `other`. So in this test, there was used even samples labeled by emotions for which this model was not trained.

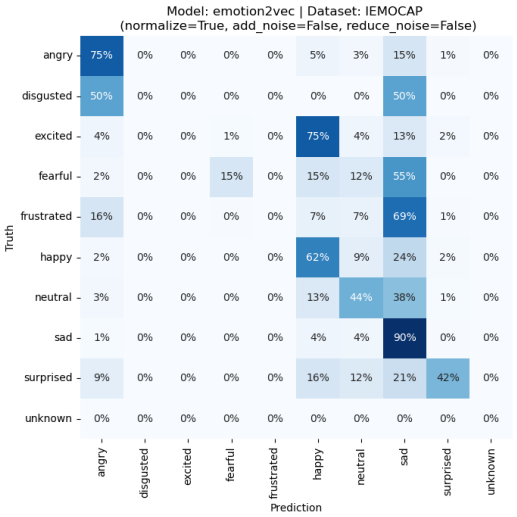


**Figure 7.6.** Confusion matrix showing how speech model Emotion2vec classified samples from the IEMOCAP dataset. This time there was used all samples from the IEMOCAP dataset, excluded were samples labeled as `other`. So in this test, there was used even samples labeled by emotions for which this model was not trained.

Now we will use the sentiment dictionary from the previous section and convert results from the last experiment presented in Figure 7.5 and in Figure 7.6 to five categories from our sentiment dictionary. Remember that these five categories are positive, negative, neutral, ambiguous, and unknown. Results of these fusions for both models we will again represent by confusion matrices. These follow.

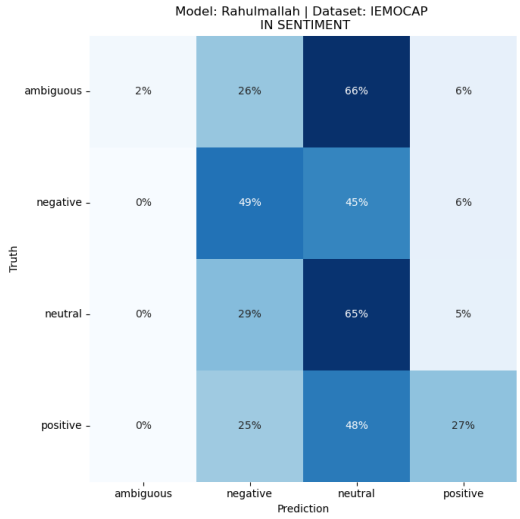


**Figure 7.7.** Confusion matrix showing how text model Rahulmallah classified all samples, excluding samples labeled other, from the IEMOCAP dataset. Same as in the previous test. But this time results as well as labels were transformed into five sentiment categories according to our sentiment dictionary.

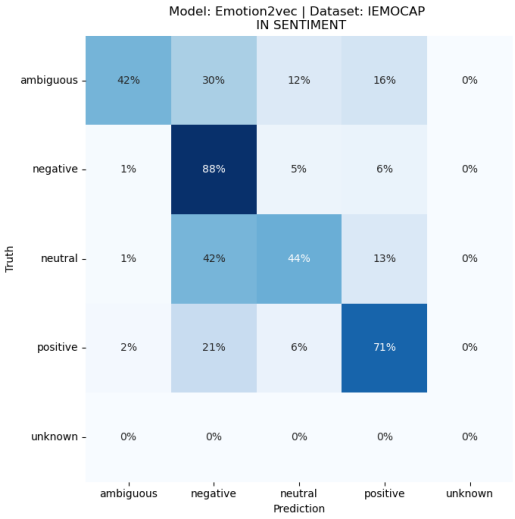


**Figure 7.8.** Confusion matrix showing how speech model Emotion2vec classified all samples, excluding samples labeled other, from the IEMOCAP dataset. Same as in the previous test. But this time results as well as labels were transformed into five sentiment categories according to our sentiment dictionary.

As you can see, the text model has a higher tendency to classify samples as neutral. It can make sense because in many situations emotions are contained only in voice and written phrases can seem neutral. The speech model is in these situations more precious.

In order to evaluate, how good the models are separately and in comparison with the output after the combination of both their results together, using our decision tree from the previous section, we have introduced two **metrics**.

- First `correct emotion`, says how many outputs contain the correct emotion corresponding to the sample label. This number we want to maximize.
- The second metric `false positive/negative` says how many outputs contain the emotion of the opposite sentiment than the label. The opposite sentiment means that the output contains emotion from the category positive, but the sample was labeled by emotion from the category negative, or vice versa. This number we want to minimize.

These metrics calculated for tests presented in Figure 7.5 and Figure 7.6 above are in the following Table 7.1. Under the line in the table, there are additional information. Row `mark as unknown`, says how many times was unknown outputted. Row `correct sentiment`, says how many times the output contained emotion from the positive category when the sample was labeled by the emotion from the positive category and vice versa. This `correct sentiment` also counts, when output is `neutral` and the sample was labeled as `neutral`.

Just to remind as we have defined before, we have five categories in our sentiment dictionary positive, negative, neutral, ambiguous, and unknown.

| metric/model: | text model | speech model |
|---|---|---|
| correct emotion | 1248 | 2991 |
| false positive/negative | 639 | 588 |
| marked as unknown | 0 | 3 |
| correct sentiment | 3578 | 5517 |
| total input samples | 7529 | 7529 |

**Table 7.1.** Tables of metrics introducing how text and speech models perform separately. Metrics were computed from the test on the IEMOCAP dataset, where all samples, excluding samples labeled as other, were used. Metrics are fully introduced in previous paragraphs.

Now finally we can test what the output of united models will look like. We will take results from separate tests for each model and use them as input for our already introduced decision tree. Our goal is to provide maximally correct and useful output. Results are presented in the following confusion matrix.

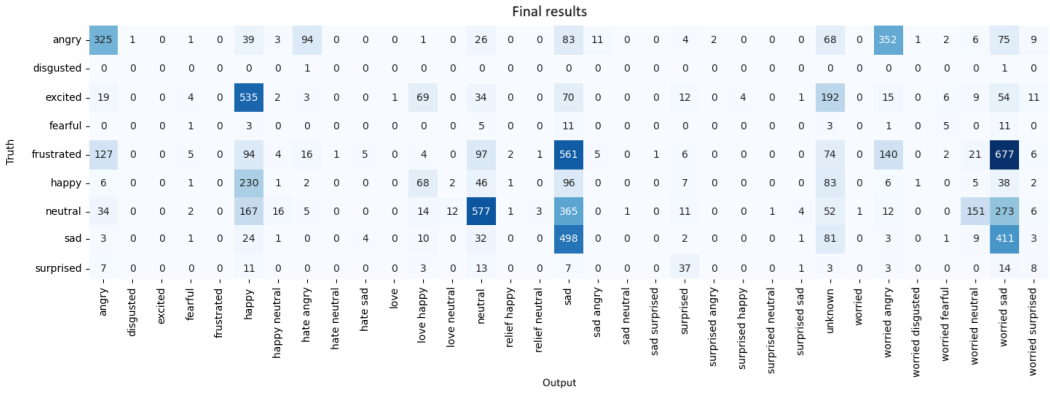


**Figure 7.9.** Confusion matrix showing final outputs of united text and speech models using the introduced decision tree. For testing was used IEMOCAP dataset in the same way as described in the previous experiment.

As you can see, most of the classifications make sense even if the models were not able to reach exactly the right emotion. For example output `worried sad` is pretty close in the way of feelings to emotion frustrated for which none of these two models was trained. As well as excited was mostly classified as happy.

To evaluate and compare this fusion of both models with their separate performances, we have added a column with calculated metrics to the previous table. You can see it whole below. Data for united models are highlighted.

| metric/model: | text model | speech model | **models united** |
|---|---|---|---|
| correct emotion | 1248 | 2991 | **2811** |
| false positive/negative | 639 | 588 | **535** |
| marked as unknown | 0 | 3 | **556** |
| correct sentiment | 3578 | 5517 | **5168** |
| total input samples | 7529 | 7529 | **7529** |

**Table 7.2.** Tables of metrics introduce how text and speech models perform separately and how the result changes when we use results from both of them to generate output using the decision tree. This column was highlighted. Tested on the IEMOCAP dataset with all samples, excluding samples labeled other. Metrics are calculated from the same data presented in the previous figure 7.9. Metrics are fully introduced in previous paragraphs.

### 7.2.4 Model fusion conclusion

Model fusion shows in the table 7.2 that by combining results from both models we can eliminate incorrectly classified emotions as emotions from opposite sentiment (`false positive/negative`), but we lose many correctly classified emotions, which are now classified as unknown. Overall we can say that by fusion we get fewer correct results but we can be more confident in their correctness.

### 7.2.5 Process chain implementation

As a result of previous research, we have implemented the final process chain. Its goal is to predict emotion from voice input only, text input only, or from combination of both.

Implementation is done in Python programming language as a class `ProcessChain` and implements previously introduced and discussed elements.

Usage of this class is very simple it has a constructor with no parameters and only one public method `recognise_emotion(wav_file_path, text_string)`, which has two optional parameters. The first of them is `wav_file_path`, we can input a path to the wav file we want to recognize an emotion from. The second parameter is `text_string` where we can input a text string containing a sentence from which we want to recognize emotions. At least one of these two parameters has to be filled.

Here in the following steps we will describe, how our algorithm works.

1. We initialise class `ProcessChain()`. Text and speech models are initialized inside.
2. Then we call the function `recognise_emotion()` with audio, text or both inputs.
3. In the method program check if at least one input is included. If not it will return an error message. Otherwise process continues.
4. If audio input is included, it checks if wav file has a sampling rate 16000 and is a mono channel. If not then the program will return an error message informing the user about this issue, otherwise process continues.
5. If only one input is included, the method processes it via a selected text or speech model and returns the resulting emotion.
6. If both inputs are included, the method processes them with selected models and sends both results into the previously introduced decision tree, which via sentiment, joins them and returns the resulting output.

The whole class is located in a file named `process_chain.py`. Below we will show the core part of the code. Three dots in the code example represent missing parts of the code.

```
class ProcessChain:
    def __init__(self):
        self._speech_model = AutoModel(
            model="iic/emotion2vec_base_finetuned",
            model_revision="v2.0.4")
        self._text_model = pipeline("text-classification",
            model="rahulmallah/autotrain-emotion-detection-1366352626")
        ...

     def recognise_emotion(self, wav_file_path=None, text_string=None):
        if wav_file_path is not None:
            wav_file = AudioSegment.from_file(
                file=wav_file_path, format="wav")

            if (wav_file.frame_rate != 16000) or
                (wav_file.channels != 1):
                print("ERROR: Input audio file
                    has to have sampling rate 16000 and 1 channel")
                print("Your file has: \n\t Sampling rate: "
                    +str(wav_file.frame_rate)+"\n\t
                    Channel number: "+str(wav_file.channels))
                return 'ERROR, wrong audio file sampling
                    rate or number of channels!'

        if wav_file_path is None and text_string is not None:
            print("Only text processing..")
            text_result = self._text_model(
                text_string, top_k=1)[0]["label"]
            text_result = self._text_model_dictionary[text_result]
            return text_result

        elif wav_file_path is not None and text_string is None:
            print("Only speech processing..")
            speech_result = self._speech_model.generate(
                wav_file_path, granularity="utterance",
                extract_embedding=False)
            speech_result = self._speech_model_dictionary[
                np.argmax(speech_result[0]['scores'])]
            return speech_result

        elif wav_file_path is None and text_string is None:
            print('ERROR: You have to provide either a wav_file_path or a
                text_string or both!')

        else:
            print("Text and speech processing..")
            speech_result = self._speech_model.generate(
                wav_file_path, granularity="utterance",
                extract_embedding=False)
```

```
            speech_result = self._speech_model_dictionary[
                np.argmax(speech_result[0]['scores'])]
            text_result = self._text_model(
                text_string, top_k=1)[0]["label"]
            text_result = self._text_model_dictionary[text_result]
            return self._decision_tree(text_result, speech_result)
    ...
```

Finally, below we will illustrate the simple use of this class with a few examples. First usage of the class in code:

```
pc = ProcessChain()
result = pc.recognise_emotion(wav_file_path=input_file_path,
    text_string=input_text)

print("Input:\n\tPath: \t"+input_file_path+"\n
    \tTranscription: \t"+input_text)
print("\nOutput:\t"+ result)
print("Ground-truth: " + label)
```

And 4 output examples:

```
1)============================================================

Input:
   Path:    ../datasets/IEMOCAP_full_release_withoutVideos/
                    IEMOCAP_full_release/Session2/sentences/
                    wav/Ses02F_impro07/Ses02F_impro07_F036.wav
   Transcription:    I will. I will. Thank you.

Text and speech processing..
rtf_avg: 0.156: 100%|      | 1/1 [00:00<00:00,  2.86it/s]

Output:   love happy
Ground-truth: excited

2)=== input wrong audio file ==================================

Input:
   Path:    ../datasets/RAVDESS/Actor_08/03-01-04-02-01-01-08.wav
ERROR: Input audio file has to have sampling rate 16000 and 1 channel
Your file has:
   Sampling rate: 48000
   Channel number: 1

Output:   ERROR, wrong audio file sampling rate or number of channels!

3)==== unknown outputted ======================================

Input:
   Path:    ../datasets/IEMOCAP_full_release_withoutVideos/
                    IEMOCAP_full_release/Session1/sentences/wav/
```

```
                    Ses01M_impro07/Ses01M_impro07_M008.wav
   Transcription:    But I don't know I want to live
                        in the housing that they give me though 'cause
                        I think I want to you know like get like
                        a nicer place.

Text and speech processing..
rtf_avg: 0.150: 100%|      | 1/1 [00:00<00:00,  1.39it/s]
  0%|          | 0/1 [00:00<?, ?it/s]
Output:   unknown
Ground-truth: excited

4)============================================================

Input:
   Path:    ../datasets/IEMOCAP_full_release_withoutVideos/
                    IEMOCAP_full_release/Session1/sentences/wav/
                    Ses01M_impro05/Ses01M_impro05_M008.wav
   Transcription:    You work here.

Text and speech processing..
rtf_avg: 0.142: 100%|      | 1/1 [00:00<00:00,  2.22it/s]
  0%|          | 0/1 [00:00<?, ?it/s]
Output:   angry
Ground-truth: angry
```

# Chapter 8
# Conclusion

First, we got to know the basics of emotions theories and how interesting but also difficult this area is, including how are emotions expressed in different cultures and that there is no only one approach to classify them.

Then we found and got familiar with many various English datasets for speech emotion recognition from which we have chosen five for our experiments. Also, we managed to get two Czech datasets, which were not publicly available before.

After we had chosen suitable datasets we successfully found a couple of usable models for the speech emotion recognition task and tested them on all seven chosen datasets. Because there is no model primarily trained for speech emotion recognition from the Czech language, which is caused also by the lack of Czech datasets, we have tried which model can best generalize for the Czech language.

While searching for models we got familiar with many platforms and portals like HuggingFace, ModelScope, and others providing easy sharing and access to various machine learning tools. We have done various tests including noise addiction and reduction, to get to know how our models behave under different conditions. To analyze the results of our tests we used apart from percentage success scores also confusion matrices.

Also, we slightly dive into emotion recognition from text and got familiar with a best-performing model.

In the end, we have taken all this knowledge, picked the best-performing model, and suggested a working process chain from input audio or text file to final emotion output.

During the whole work, we have written several Python files, classes, and methods, which were described in this work, and their full code you can find here `https://github.com/PetrStadnik/text_and_speech_emotion_recognition_bac`.

After finishing this work we already see other possibilities in continuing in this area. As the first one, because we have not found any Czech dataset for speech emotion recognition with transcription, we would like to write transcription for one, to be able to test a combination of speech and text emotion recognition in the Czech language. Another room for continuing is that, a few days before this work was completed, there was released new version of the best tested speech emotion recognition model emotion2vec+, so we would like to test it and compare it with emotion2vec tested in this work.

It was a very interesting and inspiring journey and I am already looking forward to seeing, what next we will be capable of doing with all these new machine-learning tools.

# References


[1] Charles Darwin. *The expression of the emotions in man and animals*. London: John Murray. 1st edition, 1872.

[2] Rachael E. Jack, Caroline Blais, Christoph Scheepers, Philippe G. Schyns, and Roberto Caldara. Cultural Confusions Show that Facial Expressions Are Not Universal. *Current Biology*. 2009, 19 (18), 1543-1548. DOI https://doi.org/10.1016/j.cub.2009.07.051.

[3] Wikipedia contributors. *Paul Ekman — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Paul_Ekman&oldid=1212077782`. 2024. [Online; accessed 19-April-2024.

[4] Paul Ekman. Are there basic emotions? 1992.

[5] Robert Plutchik, and Henry Kellerman. *Theories of emotion*. Academic press, 2013.

[6] Robert Plutchik. *A GENERAL PSYCHOEVOLUTIONARY THEORY OF EMOTION*. In: 1980. `https://api.semanticscholar.org/CorpusID:144721601`.

[7] Rizwan Khan. Detection of emotions from video in non-controlled environment. 2013.

[8] Andrew Ortony, and Terence Turner. What's Basic About Basic Emotions? *Psychological review*. 1990, 97 315-31. DOI 10.1037/0033-295X.97.3.315.

[9] Yi-Lin Lin, and Gang Wei. *Speech emotion recognition based on HMM and SVM*. In: *2005 international conference on machine learning and cybernetics*. 2005. 4898–4901.

[10] Hartmut Traunmüller, and Anders Eriksson. The frequency range of the voice fundamental in the speech of male and female adults. 1995, 2

[11] Jesin James, Balamurali B T, Catherine Watson, and Hansjörg Mixdorff. Exploring Prosodic Features Modelling for Secondary Emotions Needed for Empathetic Speech Synthesis. *Sensors*. 2023, 23 2999. DOI 10.3390/s23062999.

[12] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. *Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques*. 2010.

[13] *Extract MFCC*. 2024. `https://www.mathworks.com/help/audio/ref/mfcc.html`.

[14] Christian Schorkhuber, and Anssi Klapuri. *Constant-Q transform toolbox for music processing*. In: *7th sound and music computing conference, Barcelona, Spain*. 2010. 3–64.

[15] I-Tung Yang, and Handy Prayogo. Efficient Reliability Analysis of Structures Using Symbiotic Organisms Search-Based Active Learning Support Vector Machine. *Buildings*. 2022, 12 455. DOI 10.3390/buildings12040455.

[16] Hugging Face.
https://huggingface.co/docs/transformers/main_classes/feature_extractor.

[17] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. *wav2vec: Unsupervised Pre-training for Speech Recognition*. 2019.

[18] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020.

[19] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. 2021.

[20] Mashary N. Alrasheedy, Ravie Chandren Muniyandi, and Fariza Fauzi. *Text-Based Emotion Detection and Applications: A Literature Review*. In: *2022 International Conference on Cyber Resilience (ICCR)*. 2022. 1-9.

[21] Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset. *IEEE Transactions on Affective Computing*. 2014, 5 (4), 377-390. DOI 10.1109/TAFFC.2014.2336244.

[22] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*. 2008, 42 335-359. DOI 10.1007/s10579-008-9076-6.

[23] Steven R. Livingstone, and Frank A. Russo. *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)*. 2018.
https://doi.org/10.5281/zenodo.1188976.

[24] Sanaul Haq Philip Jackson. *Surrey Audio-Visual Expressed Emotion (SAVEE) Database*.
http://kahlan.eps.surrey.ac.uk/savee/.

[25] M. Kathleen Pichora-Fuller, and Kate Dupuis. *Toronto emotional speech set (TESS)*. 2020.
https://doi.org/10.5683/SP2/E8H2MF.

[26] Dominik Uhrin, Zdenka Chmelikova, Jaromir Tovarek, Pavol Partila, and Miroslav Voznak. *One approach to design of speech emotion database*. In: Misty Blowers, Jonathan Williams, and Russell D. Hall, eds. *Machine Intelligence and Bio-inspired Computation: Theory and Applications X*. SPIE, 2016. 98500B.
https://doi.org/10.1117/12.2227067.

[27] Pavol Partila. *Classification of Emotions in Human Speech*. 2016.

[28] Johannes Wagner, Andreas Triantafyllopoulos, Hagen Wierstorf, Maximilian Schmitt, Felix Burkhardt, Florian Eyben, and Bjorn W Schuller. Dawn of the Transformer Era in Speech Emotion Recognition: Closing the Valence Gap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023, 1–13.

[29] *How to use our wav2vec 2.0 model for speech emotion recognition*.
https://github.com/audeering/w2v2-how-to/blob/main/notebook.ipynb.

[30] Enrique Hernández Calabrés. *wav2vec2-lg-xlsr-en-speech-emotion-recognition (Revision 17cf17c)* . 2024.
https://huggingface.co/ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition .

[31] Jonatas Grosman. *Fine-tuned XLSR-53 large model for speech recognition in English*. https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english. 2021.

[32] Ziyang Ma, Zhisheng Zheng, Jiaxin Ye, Jinchao Li, Zhifu Gao, Shiliang Zhang, and Xie Chen. emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. *arXiv preprint arXiv:2312.15185*. 2023.

[33] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, and others. SUPERB: Speech processing Universal PERformance Benchmark. *arXiv preprint arXiv:2105.01051*. 2021.

[34] Klára Losenická. Detekce emocí z psaného textu. 2024.

[35] Tim Sainburg. *timsainb/noisereduce: v1.0*. 2019. https://doi.org/10.5281/zenodo.3243139.

[36] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*. 2020, 16 (10), e1008228.

[37] Herve Bredin, and Antoine Laurent. *End-to-end speaker segmentation for overlap-aware resegmentation*. In: *Proc. Interspeech 2021*. Brno, Czech Republic: 2021.

[38] Herve Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. *pyannote.audio: neural building blocks for speaker diarization*. In: *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*. Barcelona, Spain: 2020.

[39] Zhe Liu, Anbang Xu, Yufan Guo, Jalal Mahmud, Haibin Liu, and Rama Akkiraju. *Seemo: A Computational Approach to See Emotions*. In: 2018. 1-12.

[40] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. *GoEmotions: A Dataset of Fine-Grained Emotions*. In: *58th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020.

# Appendix A
# Attached files

| | |
|---|---|
| `read_iemocap.py` | Python class for IEMOCAP dataset loading. |
| `read_dataset.py` | Python class for loading datasets used for testing. |
| `process_wav.py` | Python class for wav files preprocessing and analyzing. |
| `analyze.py` | Python class for test results analyzing. |
| | |
| `emotion2vec.py` | Python script for emotion2vec model testing. |
| `wav2vec2.py` | Python script for ehcalabres wav2vec2 model testing. |
| `hubert.py` | Python script for s3prl hubert model testing |
| `my_w2v2.py` | Python script for my wav2vec model training and testing |
| `support_vectors.joblib` | Trained classifier on CREMA-D dataset. |
| `norm_support_vectors.joblib` | Trained classifier on normalized CREMA-D dataset. |
| | |
| `process_chain.py` | Python class including fusion of text and speech model. |