





UNIVERSITÉ DE LORRAINE

MASTER THESIS

# Evaluating the Impact of Speaker Anonymization on Emotion Recognition

*Author:*
Colleen BEAUMARD

*Supervisors:*
Hubert NOURTEL
Denis JOUVET

*A thesis submitted in fulfillment of the requirements*
*for the degree of Master of Natural Language Processing*

*in the*

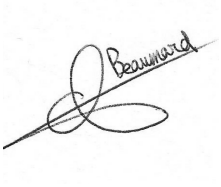
Loria - MULTISPEECH Team
Institut des sciences du Digital, Management & Cognition

August 23, 2022

# Declaration of Authorship

I, Colleen BEAUMARD, declare that this thesis titled, "Evaluating the Impact of Speaker Anonymization on Emotion Recognition" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.



Signed:

Date: 08/23/2022

*"Believe in yourself. You are braver than you think, more talented than you know, and capable of more than you imagine."*

*"Believe in your infinite potential. Your only limitations are those you set upon yourself"*

Roy T. Bennett, The Light in the Heart

UNIVERSITÉ DE LORRAINE

# *Abstract*

Institut des sciences du Digital, Management & Cognition
Management & Innovation

Master of Natural Language Processing

**Evaluating the Impact of Speaker Anonymization on Emotion Recognition**

by Colleen BEAUMARD

Speech Processing is a domain that became necessary with the dawn of tools using our voice. The drawback of those tools is our privacy: do we have to forsake them for practicality? Sensitive information can be retrieved from our voice, such as our age, our gender, etc. Speaker Recognition domain aims at removing such information by anonymizing speech signals while keeping linguistic information. However, among the information that can be used against our will, what about emotions conveyed by speech signals? Can they be considered sensitive information? Whether it is or not, we have to know if it is possible to retrieve emotion from anonymized speech signals. For this purpose, we use a Speech Emotion Recognition model, which will categorize speech signals into different emotion classes.
Hence, the aim of the Master Thesis is to build an efficient model to recognize emotion in a speech signal; the approach has been developed and evaluated on non-anonymized speech, and should later be applied also on anonymized speech.

We used SIDEKIT, an open source package created for Speaker Rocognition. We adapted it for Speech Emotion Recognition, and used one of its model, WavLM-Ecapa. By fine-tuning hyperparameters (for the loss for example) and other parameters (the speeches' training duration for example), we achieved significant results in Unweighted Average Recall (*UAR*) and Accuracy (*Acc*), two common metrics in Speech Emotion Recognition. For the best configuration, our model leads to an *UAR* of 75,3% and an *Acc* of 74,0%, which is better than the state-of-the-art by 7.3 points for *UAR* and 2.4 points for *Acc*.

Further work is necessary to know if emotion can be retrieved from anonymized speech signals, and also to experience the robustness of our model to other recording conditions.

# *Acknowledgements*

I will first talk about my coworkers.

Members of the MULTISPEECH team are just wonderful people. Each day, during five months, I was surrounded by happiness and smiles (and ping-pong). Your joy every day was really pleasing and made me enjoy my internship. I will definitely miss your company, but this not a farewell: only a goodbye. Thank you for being you.

Pierre, I want to thank you for your help with SIDEKIT and for your help throughout my internship. Your presence was really refreshing and your advice and ideas very useful.

Hubert, I want to thank you for everything. I learned a lot from you and your kindness helped me to grow both as a researcher, and as a person. Thank you for giving me a chance to show my abilities, and thank you for proofreading my thesis.

Denis, I want to thank you for your kindness, your advice throughout this internship and taking time to proofread this thesis. Thank you for giving me a chance to show my abilities for this internship, and last year's internship.

For the cafeteria members -and especially Isabelle-, your cooking was delicious. It helped me everyday by filling my heart (and my stomach) with delight.

Bruno, I want to thank you for your help during this thesis as my godfather. Your advice helped me during my internship.

Moreover, all experiments presented in this thesis were carried out using the Grid'5000[1] testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations.

Secondly, I want to talk about my Master's teachers.

Maxime, you are one of the best teachers I had during my school education. I thank you for your help, your dedication to your students, your smile, and your patience for the last two years.

Miguel, your guidance and your energy were heartwarming. This Master 2 year was difficult, but you did your best to help your students, and I thank you for that.

Ultimately, I want to thank my friends and my family.

Without my friends, I would not be here, in this Master. Thank you for believing in me, supporting me and pushing me to go as far as I can.

I want to thank my family from my deepest part of my heart. Thank you for your love and support from the day I was born. Thank you for pushing me and letting me study what I wanted to study. I know there were dark times, but I can affirm now that I have found my path.

[1] https://www.grid5000.fr

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **ASR** | Automatic Speech Recognition |
| **Acc** | Accuracy |
| **Adam** | Adaptative Moment estimation |
| **aam loss** | Additive Angular Margin loss |
| **ang** | Anger |
| **exc** | Excited |
| **fea** | Fear |
| **fru** | Frustration |
| **hap** | Happiness |
| **IEMOCAP** | Interactive Emotional Dyadic Motion Capture Database |
| **lr** | Learning rate |
| **MFCC** | Mel-Frequency Cepstrum Coefficients |
| **NLP** | Natural Language Processing |
| **neu** | Neutral |
| **xxx** | No agreement |
| **oth** | Other |
| **sad** | Sadness |
| **SR** | Speaker Recognition |
| **SER** | Speech Emotion Recognition |
| **sur** | Surprise |
| **SVM** | Support Vector Machine |
| **UAR** | Unweighted Average Recall |

# Chapter 1

# Work Environment

## 1.1 Structures

### 1.1.1 Inria

Inria is a French research institute in Digital Science and Technology that has nine centers (with eight in France and one in Chile). Three thousand nine hundred researchers and engineers are working in those centers split into two hundred and sixteen teams. There are several domains covered, such as "Digital Health, Biology and Earth", "Applied Mathematics, Computation and Simulation" or "Perception, Cognition, and Interaction". Inria is also helping the creation of startups with its "Startup Studio" program. Over two hundred startups have already been launched since 2005. It has several international associate teams and partnerships with European countries, which strengthen their scientific, technological, and societal impact.

Inria Nancy - Grand Est Research Centre center opened in 1986 and has a branch at Université de Strasbourg and in Saarbrücken. Twenty teams are split among those places. Its actual director is Bruno Lévy and it has two priorities: algorithmic intelligence and software-hardware interaction. Inria Nancy - Grand Est has also several partnerships, with companies (Orange, Thales, Studio MAIA...), with CNRS, or with DFKI (German Research Center for Artificial Intelligence) for example.

### 1.1.2 LORIA

It is a French research unit belonging to Inria Nancy - Grand Est, CNRS, and Université de Lorraine. It was created in 1997 and aims to deal with research in computer science. There are twenty-eight teams, of which fifteen are common to Inria Nancy - Grand Est. It is also part of Charles Hermite Federation, which regroups the main laboratories in Mathematics, Information and Communication Sciences, and in Control and Automation (**C**entre de **R**echerche en **A**utomatique de **N**ancy and **I**nstitut **E**lie **C**artan de **L**orraine are also parts of this federation).

LORIA's teams are splitted in five departments:

- Algorithms, Computation, Image & Geometry
- Formal methods
- Networks, Systems and Services
- Natural Language Processing & Knowledge Discovery
- Complex Systems, Artificial Intelligence and Robotics

## 1.2 MULTISPEECH team: Speech Modeling for Facilitating Oral-Based Communication


FIGURE 1.1: MULTISPEECH's logo

MULTISPEECH team is part of the Natural Language Processing & Knowledge Discovery department in LORIA, joint with Inria Nancy - Grand Est, Université de Lorraine, and CNRS. MULTISPEECH researches focus on speech processing, multisource (speech recognition), multilingual, and multimodal aspects (audiovisual synthesis). Three axes compose its research program:

- Beyond black-box supervised learning is about three challenges of deep learning
  - Integration of domain knowledge: the goal is to have models that can be reused for several tasks thanks to deep generative models and hybrid methods which combine deep learning with statistical signal processing. The use of symbolic reasoning also increases the generalization of those models.
  - Data efficiency: the majority of data are unlabeled, because of the cost of annotation, which makes model training difficult since they provide less information. Weakly supervised learning based on those data can help the noise modelisation, and can use those for unbiased training. Transfer learning is also studied to adapt an expressive speech synthesis model trained on a specific speaker to the voice of another speaker for which only neutral voice is available.
  - Privacy preservation: it aims at hiding speaker identity from speech signals by removing or altering some traits or states in the voice. Model personalization is also explored by using semi-decentralized learning.
- Speech production and perception are related to each other with articulatory modeling and multimodal expressive speech synthesis. Moreover, there are also topics related to the perception of speech (with the categorization of sounds) and prosody in both native and non-native speech
  - Articulatory modeling: 2D or 3D modeling of vocal tract dynamics is done, and the coarticulation model (developed for the animation of visible articulators) will also be extended to the face and tongue. It will help understand better the link between the face and the articulators as well as illustrate articulators' actions to learners. Different types of speech are studied, including stuttering speech.
  - Multimodal expressive speech: the goal is to improve talking heads by further developing the animations of the face's lower part (related to speech) and the face's upper part (related to facial expression). Expressivity of speech is also investigated for both parts and those talking heads are tested in various situations (with children, deaf people...).

  - Categorization of sounds and prosody: through studies of language acquisition (either with children or L2 learners), prosody is analyzed to understand its impact on speech communications. Speech technologies for computer-assisted language learning are tested in middle and high schools.

- Speech in its environment is splitted into three themes
  - Acoustic environment analyzis: it is important to analyze audio scenes in which spoken communication can happen by detecting audio events and providing semantic interpretations to them. The characterization of the environment is possible thanks to labeled/unlabeled data in real-world conditions. Estimation of room properties by room acoustics is also useful for several applications (surveillance, augmented reality...).
  - Speech enhancement and noise robustness are targeting distortions such as echo, reverberation or noise for speaker and speech recognition. Statistical signal models are explored for this aim.
  - Linguistic and semantic processing exploits word or sentence embedding which carries semantic information and combines them with acoustical uncertainty to rescore the recognizer outputs. This processing is applied to detect hate speech on social media.

This team is collaborating with international researchers (Rio de Janeiro, Stockholm, Tunis...) and companies (Facebook AI Research, Honda Research Institute (Japan), Ascora (Germany)...), as well as with national laboratories (Analyse et Traitement Informatique de la Langue Française (ATILF), Laboratoire Ligérien de Linguistique (LLL)...).

MULTISPEECH's works is applied in several domains in order to facilitate oral communication:

- Multimodal computer interactions: speech synthesis is useful for many applications (smartphones, for example) to help human-machine interaction. For example, audiovisual speech synthesis can create a talking head which benefits hard-of-hearing individuals thanks to the animated face. Expressive acoustic synthesis is also an interesting domain because of the interest of having a computer reading a story (audiobook).

- Private-by-design robust speech recognition: many speech applications are using centralized servers to process speech signals, which is dangerous for sensitive information. Having a way to process them directly on the user's terminal is a way to assure those information are not shared. However, it can affect the efficiency of the speech applications since they need a huge amount of data to be trained on. Anonymizing the speech signals before processing them on centralized servers is a solution, with respect to robustness to noise and environment.

- Aided communication and monitoring: it is the location and detection of events inside apartments, and globally locating people in a space thanks to sound detection. A foreseen application is to help disabled or elderly people, even if adapting speech recognition to elderly voices will need specific datasets. Sound monitoring can be applied to other fields (security and environmental monitoring for example).
- Computer-assisted learning: it is always difficult to learn a new language, but computers can assist us in this task. By registering a native speaker of the target language, the computer can compare the production of the learner with a native speaker and help him correct his misspronunciations.

# Chapter 2

# Context and state of the art

In this chapter, we will first present the context for Speech Emotion Recognition (*SER*) and Speaker Recognition (*SR*). We will explain what is an emotion, and why those fields are needed in our era. The second part will focus on the dataset we used for our trainings, while the third part will be about the state of the art. Finally, we will present the open package plus the model we used by detailing the last one.

## 2.1 Context

The ANR project Deep - Privacy started in 2019 and aims at developing a model capable of anonymizing a speech signal by removing everything related to the speaker's identity. Intuitively, when we hear about something related to the speaker's identity, we think about the linguistic content: for example, if someone says its name or address, or even its credit card number aloud, it is evident that those are considered as sensitive information. However, Kröger, Lutz, and Raschke, 2020 explained that, apart from those, information related to the speaker's identity can be found in different parts of the voice recording: the voice characteristics, the speech characteristics, non-speech human sounds, and background sounds. Indeed, all those information can serve as clues for the age or the mental health of the speaker for example (Figure A.1 in Appendix A identifies both sensitive information and what you can deduce from them). To protect the speaker's identity, those clues are removed from the original speech. However, in the case of an emotional speech, is the emotion still present in the anonymized speech signal?

A related question is: is the emotional state sensitive information? If yes, emotions should not be recognized in anonymized speech, but if not, they should be still recognized. In this internship, we aim at developing a model able to recognize emotions with original (non-anonymized) speech. We will focus only on the non linguistic content, the speech signal. Thereafter, this model will be applied on anonymized speech signal.

But, before viewing the emotion from Natural Language Processing (*NLP*) point of view, we have to understand what is initially an emotion.

A theory of basic emotions was proposed by psychologist named Plutchik (Contributors to Wikipedia projects, n.d.), which contains eight primary emotions (anger, fear, sadness, disgust, surprise, anticipation, trust, and joy) and ten postulates (Table A.1). Each primary emotion is the trigger to a behavior for survival (fear triggers the fight-or-flight behavior for example) and is associated to a psychological defense mechanisms (fear is danger ; anger is ennemy ; joy is possess ; sadness is isolation ; trust is friend ; disgust is poison ; anticipation is what's out there ; surprise is what is it).

Plutchik made a wheel to represent his theory, which is named "Plutchik's Wheel of Emotions" (Figure A.2). Each primary emotion is associated to a color and an emotion degree of feeling.
The neutral emotion was also added to Plutchik's theory of basic emotions to represent the non emotional state.

The aim of *SER* is to recognize emotion from speech signals. In the literature, the most considered emotions are neutral (*neu*), anger (*ang*), sadness (*sad*), fear (*fea*), disgust (*dis*), and happiness (*hap*). Others, such as frustration (*fru*) or excitement (*exc*), are present in some datasets. The lack of emotion annotated datasets is one of *SER* main issues. Indeed, this type of dataset is complicated to set up. Is it better to hire actors to play emotions, scripted or improvised, or to annotate interactions between non-actors people? Furthermore, an underlying difficulty is the annotation itself. Emotion perception is subjective, and agreement between annotators is not perfect.
From *NLP* point of view, having a model that can recognize emotions would be useful in several fields: health, to have a patient follow-up, customer services, to drive the customer to either a human or an Artificial Intelligence, or privacy (which is the main interest of this internship).

## 2.2 Interactive Emotional Dyadic Motion Capture Database

Interactive Emotional Dyadic Motion Capture Database (*IEMOCAP*) (Busso et al., 2008) is an American multimodal dataset containing audios and videos created in 2007. It is divided into five sessions where two actors (one man and one woman) acted one or several emotions in a scene. Actors were filmed, and one of them had the face covered by captors for motion capture. Since we are only interested in speech, we used only audio files. The scene was either scripted or improvised. In total, ten actors were registered (five men and five women). Each session is divided into scenes, and each scene contains several speech turns. There are 10039 speech turns in total.

Emotion annotations are both categorical (speech turns belong to a category, *neu* for example) and dimensional (speech turns are evaluated on three features: valence, arousal, and dominance).

### 2.2.1 Categorical annotation

Nine emotions were annotated : anger (*ang*), disgust (*dis*), excited (*exc*), fear (*fea*), frustration (*fru*), happiness (*hap*), neutral (*neu*), sadness (*sad*) and surprise (*sur*). The last four ones were added during the annotation process (especially for the improvised scenes). There are also two other categories: no agreement (*xxx*) and other (*oth*). The first one is when the annotators could not agree and the second is if the perceived emotion was not in the proposed categories.

For this dataset, in the literature, only four emotions are used: *neu*, *ang*, *sad* and *hap+exc* merged (Pappagari et al., 2020, Chen et al., 2021, Li, Zhao, and Kawahara, 2019). Those will be the emotion classes considered for trainings. We then used 5531 speech turns instead of 10039.

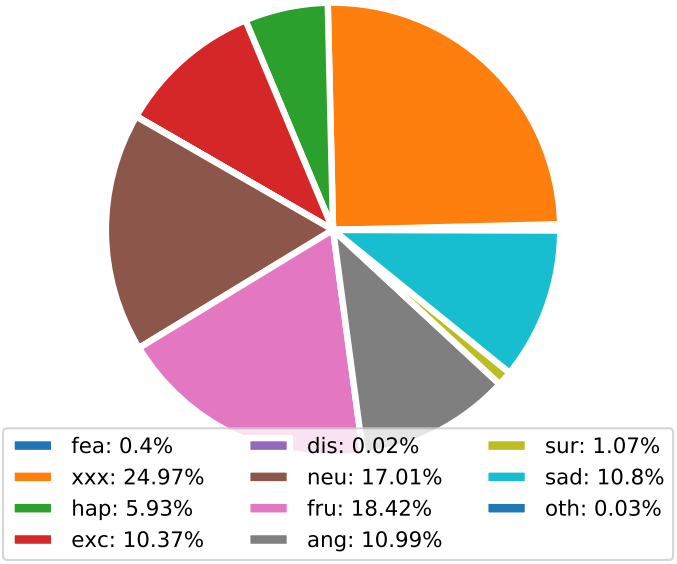



FIGURE 2.1: Distribution of all annotated emotions in IEMOCAP dataset

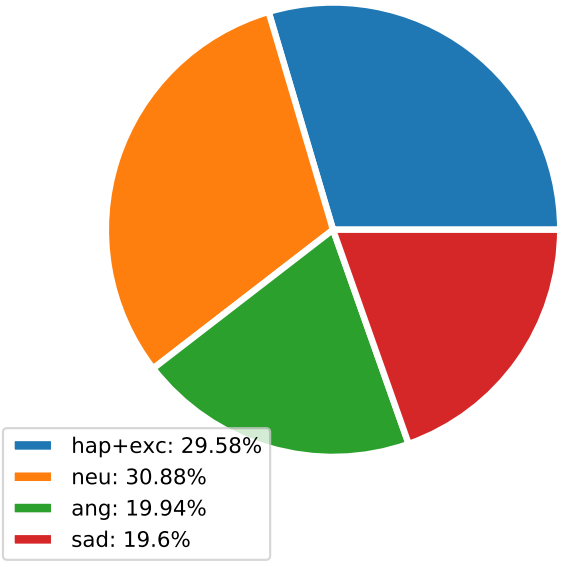



FIGURE 2.2: Distribution of the considered emotions in IEMOCAP dataset

Figure 2.1 shows the repartition of all the annotated emotions in the dataset. We can see it is not balanced: *fru*, *neu* and *ang* are over represented compared to *sur* or *hap* for example (by letting aside *xxx*).

Figure 2.2 shows the repartition of the considered emotions in the dataset. We can see it is more balanced than previously, even if *neu* and *hap+exc* are more represented than *ang* and *sad*. Each session has its own over represented emotion: *neu* for session 1, *neu* and *hap+exc* for session 2, *neu* and *sad* for session 3, *ang* for Session 4, and *neu* and *hap+exc* for session 5. Figure B.1 (in Appendix B) presents the number of speech turns for each session per emotion according to its duration. This unbalanced distribution is also present in each session: Figure B.2 presents the number of speech turns for each emotion per session according to its duration.

Figure B.3 shows us that the majority of speech turns (for both all and considered emotions) are under 15 seconds, with a peak at around 4 seconds.

### 2.2.2 Dimensional annotation

Dimensional annotation in this dataset contains three features on a scale of one to five. To represent those features, Self-Assessment Manikins pictures were used to help annotators evaluate the actors (Figure 2.3):

- **Valence** is the positivity or negativity of an emotion (first row in Figure 2.3)
- **Arousal** (or **Activation**) is the degree of reaction over the emotion (second row in Figure 2.3)
- **Dominance** is the control of the person over the emotion (third row in Figure 2.3)

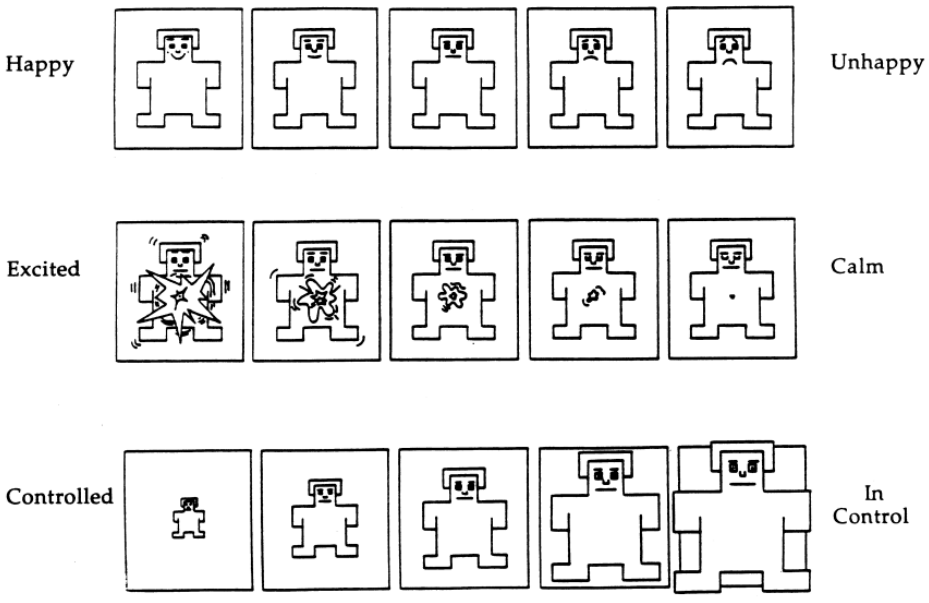



FIGURE 2.3: Self-Assessment Manikins pictures to help annotators evaluate valence, arousal and dominance

For the sake of comparability with the state-of-the-art, we will not use dimensional annotation.

### 2.2.3 Inter-annotators agreement

Three annotators annotated the entire dataset, and a fourth one (who is one of the actors for the considered session) was present for some sessions. Figure 2.4 shows Fleiss' Kappas for the annotators.

| Session | Original labels | | Recalculated labels | |
|---|---|---|---|---|
| | All turns | Reached agreement | All turns | Reached agreement |
| Entire database | 0.27 | 0.40 | 0.35 | 0.48 |
| Scripted sessions | 0.20 | 0.36 | 0.26 | 0.42 |
| Spontaneous sessions | 0.34 | 0.43 | 0.44 | 0.52 |

FIGURE 2.4: Fleiss' Kappa to measure inter-annotator agreement from (Busso et al., 2008)

The first column "Original labels" is when considering all the labels separately while the second column "Recalculated labels" is with a clustering of emotions: *hap* and *exc* were merged, *dis*, *sur* and *fea* were relabeled as *oth* (only for this evaluation). The other emotions remained the same.

Annotators had the choice to put either one or two emotion labels for each speech turn. To classify speech turns, majority voting was used by considering the single or multiple labels attributed by each annotator. The "Reached agreement" column is the speech turns for which an agreement was found through majority vote. It led to 74.60% of agreement (66.90% for scripted sessions and 83.10% for spontaneous ones) of the turns. In Figure 2.4, only those turns were used to compute Fleiss' Kappa in this column while all the turns were considered for the "All turns" column.

As expected from personal perceptions, the Fleiss' Kappa for *IEMOCAP* is either fair or moderate agreement, depending on the labels and the speech turns considered (the worst being a slight agreement for all turns for scripted sessions and the best being a moderate agreement for reached agreement turns for spontaneous sessions).

## 2.3 State of the art

*IEMOCAP* is the most used dataset for *SER*, and as explained earlier, only four emotions are considered (*neu*, *ang*, *sad*, *hap+exc*). All the sessions are used and the model is evaluated with cross-validation: four sessions are used for training while the last one is used for testing. Then, we merge the results of each sub-model (five in total) to get the model efficiency.

The metrics used to evaluate *SER* model are the Unweighted Average Recall (*UAR*) and the Accuracy (*Acc*):

- ***UAR***: $\frac{true\ positives}{true\ positives\ +\ false\ negatives} * 100$
- ***Acc***: $\frac{true\ positives}{total\ number\ of\ predictions} * 100$

We also computed for each *UAR* and *Acc* its 95% confidence interval thanks to this formula: $1.96 \times \sqrt{\frac{x*(1-x)}{n}}$ where $x$ is either *UAR* or *Acc*, and $n$ the total number of speech turns (always equals to 5531). We calculated the confidence interval for state of the art *UAR* and *Acc*, which are both ±1.0.

| **Features** | **Model** | ***UAR*(%)** | **Reference** |
|---|---|---|---|
| *MFCC* | Fined-tuned ResNet | **66.0** ±1.0 | Pappagari et al., 2020 |
| Spectrogram | *CNN* | 64.8 ±1.0 | Li, Zhao, and Kawahara, 2019 |
| *ASR* Features | *DNN* | 64.4 ±1.0 | Yeh, Lin, and Lee, 2020 |
| | | ***Acc*(%)** | |
| Raw speech | WavLM Large | **70.6** ±1.0 | Chen et al., 2021 |
| Raw speech | WavLM Base+ | 68.6 ±1.0 | |
| Raw speech | WavLM Base | 66.0 ±1.0 | |

*ASR*: Automatic Speech Recognition
*MFCC*: Mel-frequency cepstrum coefficients

TABLE 2.1: State of the art on *IEMOCAP* dataset with *UAR* and *Acc*

Table 2.1 shows the state-of-the-art for *SER* on the *IEMOCAP*.

The best *UAR* was obtain by Pappagari et al., 2020 with *MFCC* and a fined-tuned ResNet. They explore the dependencies between speaker and emotion recognition tasks by using transfer learning. Their model is specialized in speaker recognition task, and the final layer was changed for the emotion recognition task.

| Component | Layer | Output Size |
|---|---|---|
| Frame-level Representation Learning | $7 \times 7, 16$ | $T \times 23$ |
| | $\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 3$ | $T \times 23$ |
| | $\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 4$, stride 2 | $\frac{T}{2} \times 12$ |
| | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 6$, stride 2 | $\frac{T}{4} \times 6$ |
| | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$, stride 2 | $\frac{T}{8} \times 3$ |
| | average pool $1 \times 3$ | $\frac{T}{8}$ |
| Pooling | 32 heads attention | $32 \times 128$ |
| Utterance-level Classifier | FC | 400 |
| | FC | #spk:12,872 |

FIGURE 2.5: ResNet model for (Pappagari et al., 2020)

Figure 2.5 is a representation of the speaker and emotion recognition model they used (with the last layer being different for each task). There are three different parts:

1. Frame-level Representation Learning part is composed of several 2D convolutional layers with shortcuts between them;

2. Pooling part summarizes the whole utterance into large embeddings by using attention heads. Each one of them captures a different aspect of the input, and their embedding are then concatenated to form the final embedding;

3. Utterance-level Classifier part predicts the speaker for the considered speech.

The emotion recognition model is the same as the speaker recognition one, but with the last layer replaced to predict the emotions. They tested both clean *IEMOCAP* (no augmentation data) and clean + augmented *IEMOCAP*. The augmentation is when the data are augmented with noise and music from MUSAN dataset. Clean data results are better than the augmented data results when the model is pretrained on clean data while it is the opposite when pretrained on augmented data. Since we did not planned to use data augmentation, we will consider only the clean *IEMOCAP* results with the model pretrained on clean data.

Pappagari et al., 2020 achieved 66.0% *UAR* by fine-tuning their model and using *MFCC* as speech features.

The best *Acc* was obtain by Chen et al., 2021 by using raw speech and WavLM Large model. Their goal was to use self-supervised learning for different speech tasks.

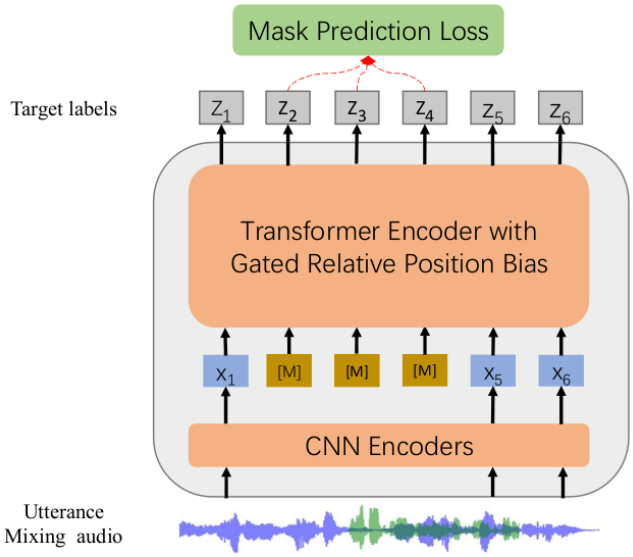



FIGURE 2.6: WavLM model for (Chen et al., 2021)

Figure 2.6 is a representation of the WavLM model they used. There are three different parts:

1. Convolutional Neural Network encoder part contains seven blocks of temporal convolution, and are followed by layer normalization and Gaussian Error Linear Unit (*GELU*) activation layer. The outputs are 25 ms of speech, and some of them are masked for the next part;

2. Transformer encoder part is equipped with a convolution based relative position embedding layer, and a Gated Related Position Bias. It allows to take the position of the inputs into consideration when reconstructing the speech;

3. Mask prediction loss part simulates either noise or overlapping speech by adding noise to the masked input, or overlapping the masked input with a random selected speech. Then, it predicts its label.

(Chen et al., 2021) tested three variations of WavLM model:

- WavLM Base has twelve encoded layers, 768-dimensional hidden states and eight attention heads (94.07M parameters in total). It was pre-trained on LibriSpeech for 400K steps;

- WavLM Base+ is the same as WavLM Base, but was pre-trained on 94K large-scale data (LibriLight, VoxPopuli, and GigaSpeech datasets) for 1M;

- WavLM Large has 24 encoded layers, 1027-dimensional hidden states and 12 attention heads (316.6M parameters in total). It was pre-trained on the same data as WavLM Base+ for 700k steps.

WavLM Large is the one which performed better with 70.6% of *Acc*. Unlike Chen et al., 2021, we will not augment our data with noise or overlap speech turns as explained in the third part of Chen et al., 2021's model explanation.

## 2.4 SIDEKIT

SIDEKIT (Larcher, Lee, and Meignier, 2016) is an open source package for Speaker and Language recognition. It was developed by Anthony Larcher (LST - Le Mans), Sylvain Meignier (LST - Le Mans), Kong Aik Lee (A*STAR-Singapore), Gaël Le Lan (Orange Labs), Pierre Champion (INRIA-Nancy), Florent Desnous, Ambuj Mehrish (LST -Le Mans) and Hubert Nourtel (INRIA-Nancy).

It contains tools to extract acoustic features (MFCC, LFCC...), modeling and classification models (GMM, SVM...), and tools to present the results (DET plot...). Librispeech and VoxCeleb 1 and 2 datasets were used to train several Speaker Recognition (*SE*) models.

SIDEKIT aims at recognizing the speaker with different deep neural networks models, such as WavLMEcapa-TDNN or HalfResNet34. It also proposes a custom model in which you can either create a new one or modify an already existing deep neural network.

## 2.5 Model structure

For all experiments, we used a customized version of WavLMEcapa-TDNN model for Speech Emotion Recognition (*SER*). It is the combination of two different models: WavLM and Ecapa-TDNN. We used Additive Angular Margin (*aam*) loss for the last layer of the model, and tested different schedulers. PyTorch was used for the implementation of the whole model.

Speech turns are gathered inside a batch of variable size (which is a parameter). For computational issues, the number of speech turns per category is equal to $\frac{1}{4}$ of the batch size: if the batch size is 200, there will be 50 speech turns from each emotion class. The last batch will not have equal number of speech turns for each class, but it will stay in the same order of magnitude (13, 10, 17, 9 for example). This is because each speech turn is seen once per epoch. The entirety of the batch is seen during one epoch.
Since we did cross-validation, the training and the test sets were already determined. The validation set was taken from the training set, and corresponds to 2% of its size. After each epoch, the validation set is used in order to evaluate each epoch model. The first epoch model is considered as the best one and is saved. Then, it is compared with each new epoch's model: if the newer model had an *UAR* above the best model one, it replaced the previous best model (determined thanks to the validation set). It goes until there are no more epochs.

### 2.5.1 WavLM

WavLM Large model from (Chen et al., 2021) was used for the preprocessor layer. It was loaded thanks to Self-Supervised Speech Pre-training and Representation Learning[1] (s3prl) toolkit, which regroups several models for different speech tasks.

The model was explained in the state of the art section (Figure 2.6) but contrary to (Chen et al., 2021) the last layer was not used. After speech turns were preprocessed and transformed into embeddings, they were output to the sequence network: the Ecapa-TDNN.

### 2.5.2 Ecapa-TDNN

Ecapa-TDNN model from (Desplanques, Thienpondt, and Demuynck, 2020) was used for the sequence network layer.

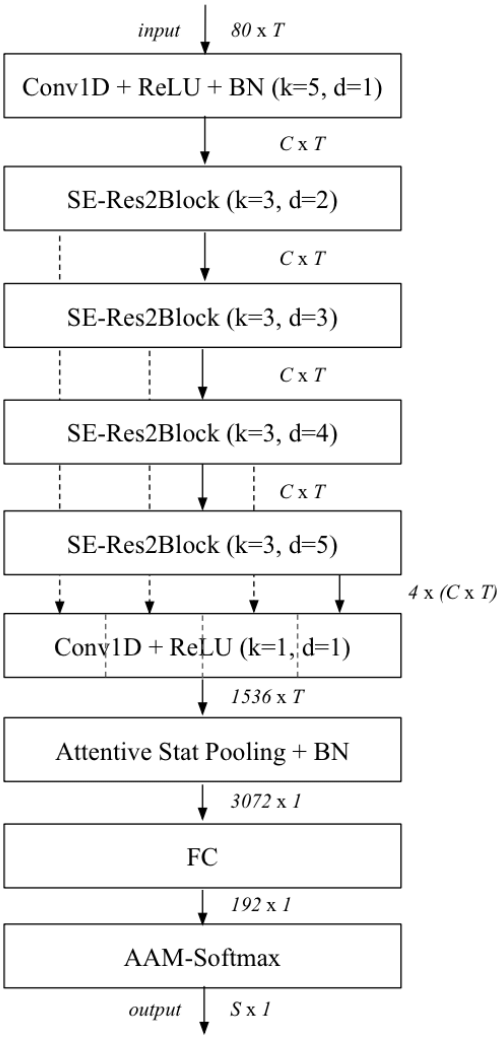



FIGURE 2.7: Ecapa-TDNN model from (Desplanques, Thienpondt, and Demuynck, 2020)

*T*: number of vectors *C*: number of values *S*: number of classes

[1] https://github.com/s3prl/s3prl

Figure 2.7 represents the Ecapa-TDNN model we used for our model. It takes as input the speech turns embeddings from the preprocessor (WavLM). The sequence network layer will aggregate the information inside the embeddings to output the most significant information to predict the speech turns classes. The final vector, called x-vector, will contain only relevant information about the speech. It is then processed through an Attention Stat Pooling layer, a Fully Connected (*FC*) layer and finally, the loss (which is the Additive Angular Margin *aam* loss).

### 2.5.3 Additive Angular Margin Loss

Choosing a loss is very important when we want to train a model: it helps to see if the model has a high rate of errors or not. The more the model is performant, the closer the loss will be to 0. The goal is to have a model which makes no errors at all and consequently to have a loss at 0.

In Deng et al., 2019, the *aam* loss (*ArcFace*) was introduced and tested for the first time. It compared it with previously used losses: Softmax cross-entropy loss, A-Softmax (*SphereFace*) and Large Margin Cosine (*CosFace*).

- Softmax cross-entropy loss is a Softmax activation plus a cross-entropy loss as the output layer
- A-Softmax (*SphereFace*) loss is a softmax loss with an angular margin hyperparameter $m$ which increases inter-class distance
- Large Margin Cosine (*CosFace*) loss is like SphereFace but the angular margin hyperparameter $m$ is substracted to the cosine angle of the features vector
- *aam* (*ArcFace*) is a softmax loss with $m$ (as *SphereFace*) and a feature scale ($s$) hyperparameter which is the radius of an hypersphere created from the center of each class

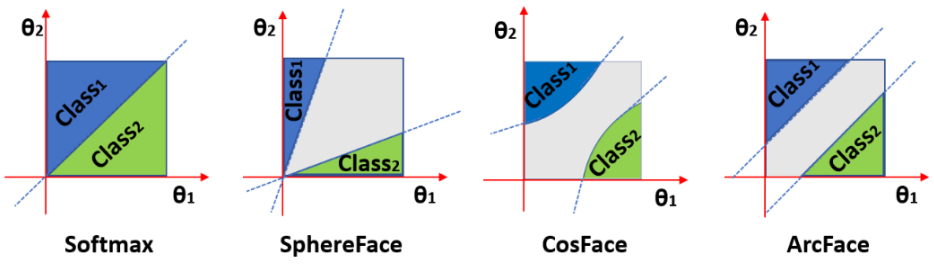



FIGURE 2.8: Schematic representation of class separation for different losses (Deng et al., 2019)

As we can see in Figure 2.8, there is a clear difference for each loss for decision boundaries: softmax activation plus cross-entropy loss is most simple while the others are more precise. The choice of the loss then depends on the data we have in order to get the most efficient model.

SIDEKIT is using *aam* (*ArcFace*) loss for speaker recognition because of its linear decision boundaries. Since a speaker cannot overlap with an other speaker, only *CosFace* and *ArcFace* losses ensure a sufficient gap between speaker classes. Finally, *aam* (*ArcFace*) is the best choice because of its linearity. *SER* being close to *SE*, we used the same loss for our model.

### Additive Angular Margin loss

This loss corresponds to the first step over the two of our model loss:

1. *aam* loss have two important hyperparameters: the margin $m$ and the feature scale $s$ (Figure 2.9 shows a schematic explanation)

   (a) $m$ is the angle value (in orange) which separates each emotion class

   (b) $s$ is the radius value of the hypersphere (in pink) created from the center of each class

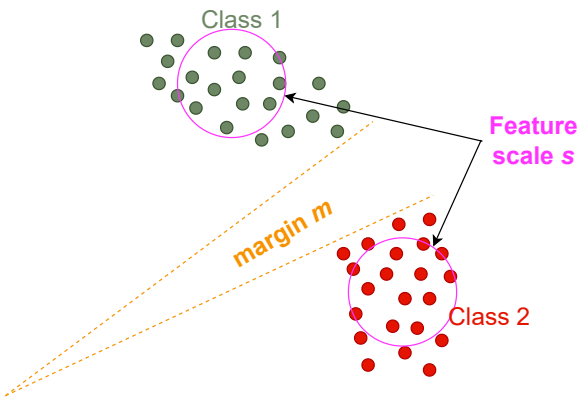



FIGURE 2.9: Schema of margin and feature scale hyperparameters for aam loss

   First, a $L_2$ normalization is done on the input, and the cosine distance is calculated between the normalized input and its target according to $m$. The output is then multiplied by $s$ to select the speech turns inside the hypersphere

2. A cross-entropy loss. In our case, the inputs are both a batch of speech turns with their probability to belong to each emotion (the output of the precedent step), and their associated targets. The output is the performance of the model: lower it is, better is the model

### 2.5.4 Adaptative Moment estimation optimizer and schedulers

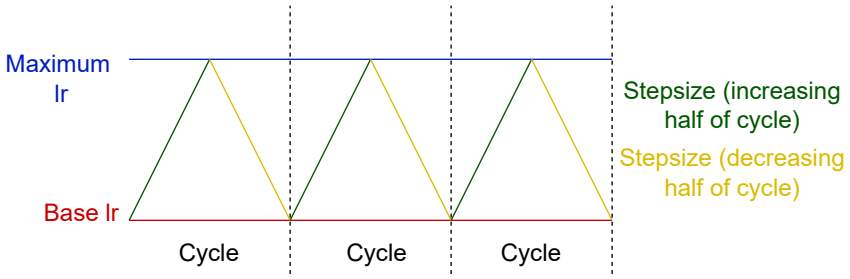
Optimizer and scheduler are important algorithms in a model. The first one helps the model to minimize the loss function while the second is here to adjust the learning rate on either each epoch or batch (depending on the chosen scheduler).

We used Adaptative Moment estimation (*Adam*) optimizer, one of the most popular. It computes learning rates for each parameter by storing the average of past gradients. This optimizer is easy to implement, computationally efficient and needs only a little amount of memory (it was already implemented within SIDEKIT).

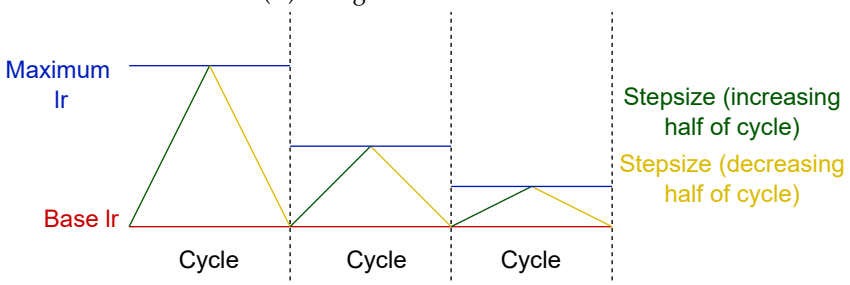
Concerning the scheduler, we tested three different ones already implemented within SIDEKIT, each with their own specificities.

#### CyclicLR

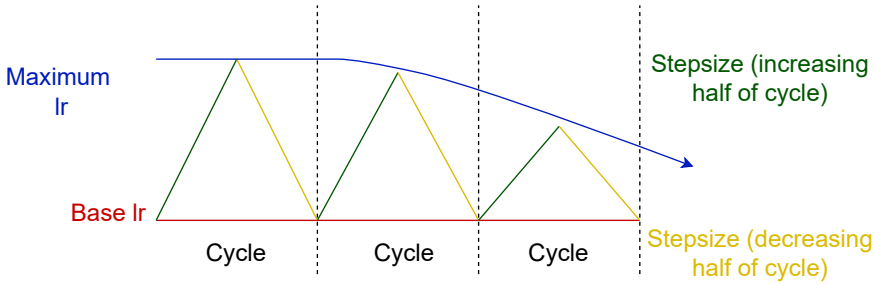
Smith, 2017 introduced CyclicLR scheduler. The idea is to let the learning rate *lr* vary between a base and a maximum *l*r (two needed parameters) within a stepsize, which is the number of iterations.




(A) triangular mode




(B) triangular2 mode




(C) exp_range mode

FIGURE 2.10: CyclicLR scheduler diagrams

Figure 2.10 contains three diagram representing each one a mode possible for this scheduler. You can specify either only the increasing stepsize for the increasing half (in that case, the decreasing stepsize is equals to the increasing stepsize), or both increasing and decreasing stepsizes. The *lr* will never goes under the base *lr*, and will never exceeds the maximum *lr*. The increasing line stops either when the maximum *lr* or the increasing stepsize is reached. It is the same for the decreasing line, it stops either when the base *lr* or the decreasing stepsize is reached.

There are three different modes[2]:

- triangular is the simplest one (represented in Figure 2.10a). The base and maximum *lr*s never change for the whole training process. Thanks to the *lr* oscillations, we can break out of saddle points more easily
- triangular2 (represented in Figure 2.10b) divides the maximum *lr* by two after each cycle. It helps decreasing the *lr* into more lower areas of *lr* landscape
- exp_range (represented in Figure 2.10c) is similar to the triangular2 mode but instead of dividing the maximum *lr* by two, it divides it by a factor of $gamma^{iterations}$ (parameter) with *iterations* being equal to the number of increasing and decreasing stepsizes. Lowering the maximum *lr* with exp_range mode allows a more finetuned control over the declining rate than with triangular2 mode

### OneCycleLR

Smith and Topin, 2019 presents OneCycleLR scheduler. Instead of doing multiple cycles like CyclicLR scheduler, OneCycleLR is doing only one cycle. The number of epochs for this cycle has to be lower than the actual epochs number to allow a decaying phase, which will lower the *lr* under the minimum *lr* given.

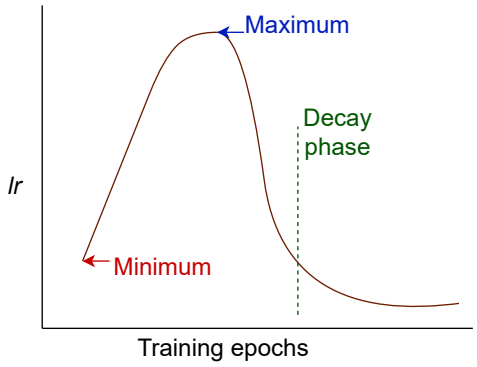



FIGURE 2.11: OneCycleLR diagram

Figure 2.11 is a representation of OneCycleLR scheduler. Both this scheduler and CyclicLR have to be stepped after each batch in order to change the *lr*, contrary to other schedulers which have to be stepped after each epoch. Several parameters can be specified, such as the maximum *lr*, the number of steps of the cycle spent increasing *lr*... However, it is mandatory to at least specify either total_steps (the total number of steps in the cycle) or the number of epochs and the number of steps per epoch.

[2]More information about each mode in: https://pyimagesearch.com/2019/07/29/cyclical-learning-rates-with-keras-and-deep-learning/

**ReduceLROnPlateau**

ReduceLROnPlateau scheduler idea is to chose a metric (generally loss) and to see if it stagnates or not for a precise number of epochs (patience parameter). When the metric began to stagnate or increase/decrease (according to which metric was chosen), the number of epochs are counted until the metric increases or decreases. If the number of epoch became greater than the patience parameter, this scheduler will reduce the *lr* by a factor you can set (parameter). A minimum *lr* can also be specified to avoid to have a very low *lr*.

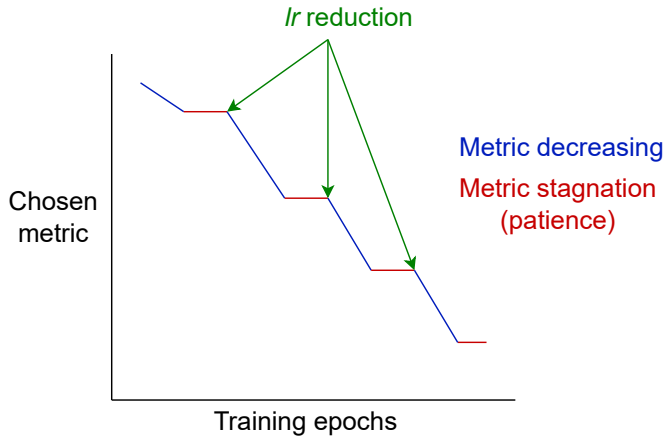



FIGURE 2.12: ReduceLROnPlateau diagrams

Figure 2.12 represents this scheduler. Since you can chose any metric, it is possible to, for example, follows the evolution of *Acc*. Mode parameter let you decide wether you want the chosen metric to increase (max mode) or to decrease (min mode). Morover, ReduceLROnPlateau uses a threshold to focus on significant changes only that you can change. The way threshold is computed can also be changed between rel ($best * (1 \pm threshold)$) and abs ($best \pm threshold$).

Contrary to CyclicLR and OneCycleLR schedulers, ReduceLROnPlateau is stepped after each epoch.

# Chapter 3

# Experiments

In this thesis, we aim at having an effective *SER* model which can recognize emotions from non-anonymized speech signals. This will be the first step of recognizing emotions from anonymized speech signals. Here, we are dealing with original (non-anonymized) speech signals.

We focused on one particular model (WavLM-Ecapa) from SIDEKIT, and did several experiments to find the best hyperparameters among the loss margin $m$ and feature scale $s$, the scheduler, the batch size, the duration of the input speech for training and testing, and finally, the learning rate $lr$.

We also experimented on annotators by selecting only some annotations. The goal was to see if, despite the low agreement between the three annotators, selecting only one or two (or even the three without the agreement) of them would give better results than considering the agreement.

## Preliminary information

For all experiments, I used four emotions categories: *neu*, *ang*, *sad* and *hap+exc* similarly to in the literature (Pappagari et al., 2020, Chen et al., 2021, Li, Zhao, and Kawahara, 2019). They will be referenced as "the 4" or "the 4 emotions" and the total number of speech turns is then 5531.
We evaluated models with cross-validation: four sessions were used for training and validation sets, and the last session was used as the test set. To have the results for all the models, we retrieved each predictions and each gold annotations. They are then concatenated before plotting the confusion matrix by comparing the predictions and the gold annotations.

SIDEKIT was adapted from its original purpose (*SE*) for *SER*. All the modifications are available on SIDEKIT GitLab page[1] under the branch "emotion".
We mostly used this package for *SER* (single-task training), but we also tested *SER* and *SR* at the same time (multi-task training). Given the results, we didn't further it.

We have decided to set the $lr$ to $10^{-4}$ to follow the experiments conducted by Chen et al., 2021 experiments, and finetuned hyperparameters and SIDEKIT parameters to find the best model for emotion recognition. Finally, we have decided to set the number of epochs to 15 giving our first results.

[1] https://git-lium.univ-lemans.fr/speaker/sidekit

## 3.1 Experiments on annotations

In this section, we first analyzed the annotations by considering two different sets of annotation and by calculating the Fleiss' Kappa for each annotator. Secondly, we compared several models trained on different annotators.

### 3.1.1 Annotations analyzis

Since emotions are personal perceptions, we analyzed the annotations provided by the various annotators. We did not considered the fourth annotator since some sessions were only annotated by the three main ones. In this part, we will refer to two different sets:

- *full agreement* will be the annotations where at least two of the three annotators agreed on an emotion within the 4 emotions considered. If an annotator used two emotions, the first one was kept if it is within the 4, otherwise the second emotion was kept if it is within the 4. If none of them were within the 4, the first one was kept (Figure C.1 in Appendix C summarizes this choice). In total, there are 5531 speech turns.
- *strong agreement* will be the annotations where both the agreement and all annotators' annotations are within the 4 emotions considered. In the case where one annotator used two emotions, we followed the same scheme explained previously but if none of them were within the 4, we did not considered the speech turn at all. In total, there are 3835 speech turns.

| **Annotator** | ***neu*** | ***ang*** | ***sad*** | ***hap+exc*** | **Total** |
|---|---|---|---|---|---|
| Annotator 1 | 575 | 1271 | 1252 | 3011 | 6109 |
| Annotator 2 | 2028 | 2116 | 1617 | 2489 | 8250 |
| Annotator 3 | 3803 | 976 | 815 | 1692 | 7286 |
| Agreement | 1708 | 1103 | 1084 | 1636 | 5531 |

TABLE 3.1: Distribution of emotions for the three annotators and the full agreement

| **Annotator** | ***neu*** | ***ang*** | ***sad*** | ***hap+exc*** | **Total** |
|---|---|---|---|---|---|
| Annotator 1 | 468 | 613 | 885 | 1869 | 3835 |
| Annotator 2 | 989 | 566 | 804 | 1476 | |
| Annotator 3 | 1492 | 526 | 559 | 1258 | |
| Agreement | 985 | 562 | 798 | 1490 | |

TABLE 3.2: Distribution of emotions for the three annotators and the strong agreement

Tables 3.1 and 3.2 are respectively showing the distribution of each emotion for the three annotators and the full agreement, and for the three annotators and the strong agreement.
As we can see in Table 3.1, the distribution of the emotions is unbalanced for each annotator. This analyzis shows to which extent the annotation of emotion is difficult and the necessity to have several annotators in order to have an agreement as close as reality as possible.

Concerning Table 3.2, we can see annotator 2's distribution of emotions is the closest to the strong agreement while annotator 1's distribution of emotions is the farthest.

| **Annotator** | **#annotations within the 4** | **#annotations outside the 4** | ***UAR* (%)** | ***Acc* (%)** |
|---|---|---|---|---|
| Annotator 1 | 4496 | 1035 | 73.3 ±1.0 | 70.8 ±1.0 |
| Annotator 2 | 5375 | 156 | **94.0** ±1.0 | **93.8** ±1.0 |
| Annotator 3 | 5026 | 505 | 78.2 ±1.0 | 81.0 ±1.0 |

TABLE 3.3: Distribution, *UAR* and *Acc* of each annotator's annotations for the full agreement set

Table 3.3 displays the number of annotations categorized inside the 4 emotions (second column) and outside of it (third column) by comparison to the full agreement for the three annotators. Each annotator's *UAR* and *Acc* was computed based on the full agreement (only for speech turns categorized as one of the 4 emotions). It confirms the annotator 2 as being the closest to the full agreement with only 156 speech turns out of the 4 emotions considered over 5531 speech turns. Moreover, his/her *UAR* and *Acc* are the higher ones among all the annotators.

| **Annotator** | **Fleiss' Kappa** |
|---|---|
| Annotator 1 | 0.79 |
| Annotator 2 | 0.93 |
| Annotator 3 | 0.81 |

TABLE 3.4: Fleiss' Kappa score for each annotator according to the strong agreement set

Table 3.4 displays the Fleiss' Kappa for each annotator according to the strong agreement set. We can see that annotators 2 and 3 obtained the higher Fleiss' Kappa (almost perfect agreement), and annotator 1 obtained the lower one (substantial agreement). These results corroborate the assessment that annotator 2 is the one closest to the strong agreement, and consequently to the full agreement.

### 3.1.2 Annotators' models

After analyzing the annotations, we have decided to make a comparative study on models trained on each annotator. Each annotator's model will be trained on this annotator's annotations for speech turns categorized inside the 4 emotions. This means that each model will be trained on a different amount of speech turns. We also trained a model on the full agreement set.

We will refer to each annotator by their number as before (1, 2 or 3). Moreover, we considered for one model the use of the double annotations (*neu-ang* for example). This specific model is written with (2 ann.). For the models trained on one annotator, its annotations were considered as the target for the loss and they were trained on their own annotation sets (see Table 3.1).

Finally, for the models trained on several annotators, we calculated each annotators cross-entropy (with their own prediction and their own target) after each prediction before summing them to produce the loss. The final prediction was the emotion with the highest probability over all the predictions. The idea behind it is that if the model is able to predict the correct target for each annotator before summing the cross-entropies, the loss will be lower than if the model is not able to do it. A lower loss means the model is capable of predicting the correct emotion for each annotator. Those models have been trained on the strong agreement (see Table 3.2).
We also divided by two the loss contribution corresponding to the annotator 1 for one model to see the impact on the model performance. We did that for this specific annotator because it is the one with the lower Fleiss' Kappa.

| Ref | Annotations | *UAR* (%) | *Acc* (%) |
|---|---|---|---|
| a | Annotator 1 | 61.6 ±1.0 | 72.2 ±1.0 |
| b | Annotator 2 | 64.8 ±1.0 | 64.8 ±1.0 |
| c | Annotator 3 | 63.3 ±1.0 | 65.2 ±1.0 |
| d | Agreement | **73.4** ±1.0 | **72.0** ±1.0 |
| e | 1, 2, 3 | 31.2 ±1.0 | 57.4 ±1.0 |
| f | 1*0.5, 2, 3 | 29.2 ±1.0 | 56.0 ±1.0 |
| g | 1, 2, 3, agreement | 31.2 ±1.0 | 56.3 ±1.0 |
| h | 1(2 ann.), 2(2 ann.), 3(2 ann.), agreement | 28.5 ±1.0 | 58.2 ±1.0 |
| i | 2, 3 | 30.8 ±1.0 | 55.0 ±1.0 |

Batch size=200 OneCycleLR(base $lr$=0.1)
Duration(train=3 secs) Duration(test=whole)
$lr$= $10^{-4}$ $m$=0.2 $s$=30

TABLE 3.5: UAR and Acc of each annotator's model and the agreement model according to their own annotation sets (first part) and the strong agreement (second part)

Results displayed in Table 3.5 showed that when you want to annotate subjective perceptions, it is better to have one agreement between several annotators.

The *UAR* provided by each annotator's model is ten points lower than the agreement model *UAR*. However, when we combined each annotator's annotation, the *UAR* was divided by about two compared to the agreement.

Model (g) has the same *UAR* but a lower *Acc* than model (e), which signifies the second model performs slightly better for predicting each class emotion while performing the same for global predictions.

Model (e) performed better than the model (f) and slightly better than the model (i). Adding the annotator 1's annotations helps to increase a bit *Acc* (57.4%, 56.0% and 55.0%), but decreases the *UAR* if its cross-entropy is divided by two (29.2% for model (f) instead of 31.2% for model (e) 30.8% for model (i)).

Model (h) performed less than the model (g) model. It can be explained by the diversity of the annotations (from minimum three up to maximum six for one speech turn), which made the learning difficult for the model.

In conclusion and according to these results, we have decided to use the agreement set for our next experiments. We will have 5531 speech turns, with 1708 *neu*, 1103 *ang*, 1084 *sad* and 1636 *hap+exc*.

## 3.2 Hyperparameters finetuning

In this section, we primarily studied the impact of two parameters of Additive Angular Margin (*aam*) loss: the margin $m$, and the feature scale $s$. Finally, we studied the impact of the scheduler on our model, and after selecting the most performant, we finetuned this scheduler principal parameter.

### 3.2.1 Impact of Additive Angular Margin loss hyperparameters

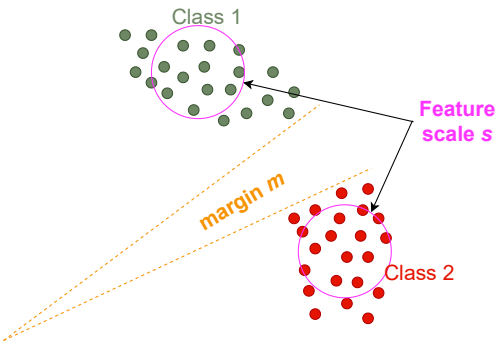



FIGURE 3.1: Schema of $m$ and $s$ hyperparameters for the *aam* loss

| *m* | *s* | *UAR*(%) | *Acc*(%) |
|---|---|---|---|
| 0.5 | 30 | 68.7 ±1.0 | 66.5 ±1.0 |
| 0.2 | | 70.1 ±1.0 | 67.9 ±1.0 |
| 0.1 | | 70.6 ±1.0 | 68.6 ±1.0 |
| | 10 | **73.4** ±1.0 | **72.0** ±1.0 |
| | 5 | 69.7 ±1.0 | 68.4 ±1.0 |

TABLE 3.6: Experiments to study the impact of *aam* loss hyperparameters

Batch size=200
OneCycleLR(base $lr$=0.1)
Duration(train=3 secs)
Duration(test=whole)
$lr$= $10^{-4}$

In this section, we wanted to test which values of both margin $m$ and feature scale $s$ hyperparameters of the *aam* loss would give the best results. In the literature, $s$ is always settled to 30 while $m$ is different for each experiment (Wang et al., 2018). Figure 3.1 reminds what are $m$ and $s$ hyperparameters.

- $m$ is the angle value (in orange) which separates each emotion class
- $s$ is the radius value of the hypersphere (in pink) created from the center of each class

Table 3.6 displays the results: the couple $m$=0.1/$s$=10 is the one giving the best performance among the couples tested. It means different emotions features are closer than what we could think a priori, and each emotion speech turns are very close to each other. This is confirmed by the escalation of the results from $m$=0.5 to $m$=0.1: the smaller $m$ is, the better the results are because the separation between the emotions are closer to reality.

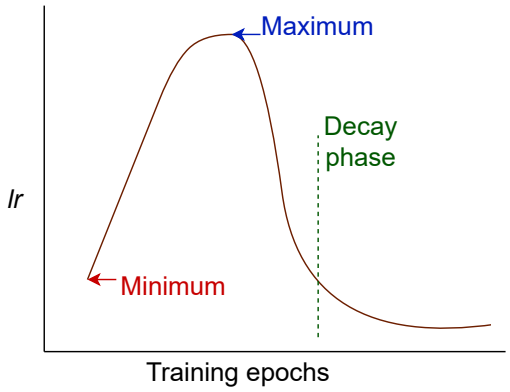
Moreover, for $m$=0.1, $s$=10 gives better results than $s$=30, which can be explained by the proximity of classes between them. If both hyperspheres are in the border of *neu* and *ang* for example, $s$=30 will take into account both hyperspheres for training for each emotion, while s=10 will only take into account each emotion hypersphere for their own training.

With this explanation, we reduced the $s$ hyperparameter to 5 and expected to have better results than $s$=10. It appeared that they were lower: we can deduce that when $s$ is too low, some hypersphere features are missed if they are far from the others (within a radius of 5 for all hyperspheres).

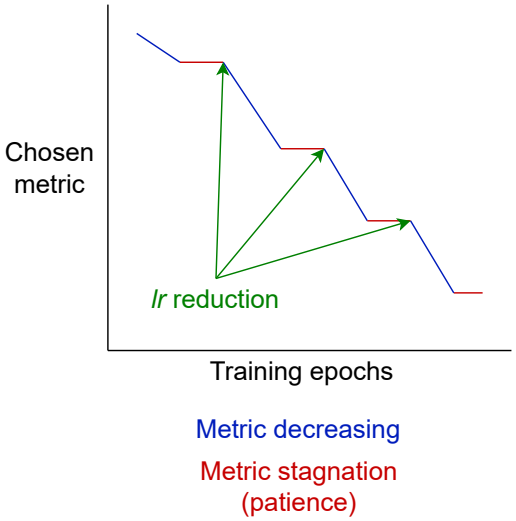
Thanks to those results, we understand why $m$ and $s$ hyperparameters are important to finetune for emotions, and more globally, for annotations that are in the domain of personal perception.

### 3.2.2 Impact of the scheduler

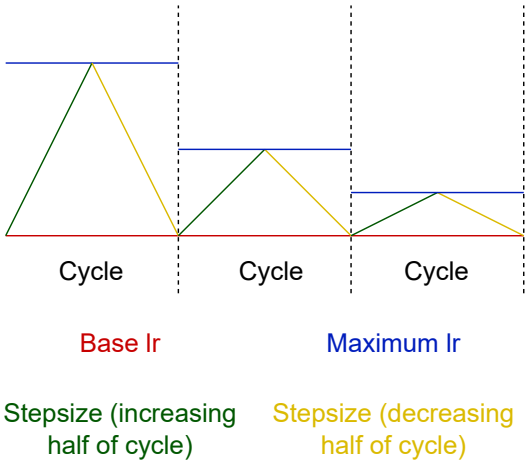
After seeing the impact of the loss hyperparameters, we wanted to study the impact of the scheduler on the results. In the previous experiments, we used OneCycleLR scheduler with a base *lr* of 0.1. Two other schedulers have been tested: CyclicLR (triangular2 mode) and ReduceLROnPlateau (we set the validation loss as the followed metric).




(A) OneCycleLR




(B) ReduceLROnPlateau




(C) CyclicLR (triangular2 mode)

FIGURE 3.2: The different tested schedulers

Figure 3.2 represents the three tested schedulers: OneCycleLR (Figure 3.2a), CyclicLR in triangular2 mode (Figure 3.2c) and ReduceLROnPlateau (Figure 3.2b).

- For OneCycleLR, we put the maximum *lr* at 0.1
- For CyclicLR, we put the base *lr* at $10^{-5}$ and the maximum *lr* at value ($10^{-4}$)
- For ReduceLROnPlateau, we tested several values of patience (5, 3, 1) and did not put any minimum *lr*

| **Scheduler** | **Patience** | ***UAR*(%)** | ***Acc*(%)** |
|---|---|---|---|
| OneCycleLR | NA | **73.4** ±1.0 | **72.0** ±1.0 |
| CyclicLR | NA | 70.0 ±1.0 | 68.0 ±1.0 |
| ReduceLROnPlateau | 1 | 70.3 ±1.0 | 68.6 ±1.0 |
| ReduceLROnPlateau | 3 | 71.1 ±1.0 | 69.5 ±1.0 |
| ReduceLROnPlateau | 5 | 71.0 ±1.0 | 69.3 ±1.0 |

Batch size=200 Duration(train=3 secs)
Duration(test=whole) $lr$=$10^{-4}$ $m$=0.1 $s$=10

TABLE 3.7: Impact of the scheduler

Table 3.7 displays the results obtained with different schedulers. We can see our actual scheduler (OneCycleLR) outperforms the others. CyclicLR perfomed the worst, which can be explained by the number of cycles and the maximum *lr* given. About ReduceLROnPlateau, we can see that patience at 3 gave the best result among patience at 1 and at 5, even if the last value has very close results to patience at 3. Waiting several epochs before lowering the *lr* seems to help the model improve itself. Since we trained our models on 15 epochs, there is a possibility those models did not have time to converge efficiently, but we lacked time to reiterate the trainings with 100 epochs.
After seeing those results, we tested several values for OneCycleLR parameter (base *lr*).

| **Base *lr*** | ***UAR*(%)** | ***Acc*(%)** |
|---|---|---|
| 0.1 | **73.4** ±1.0 | **72.0** ±1.0 |
| 0.01 | 71.9 ±1.0 | 70.5 ±1.0 |
| 0.001 | 71.3 ±1.0 | 70.0 ±1.0 |

Batch size=200 Duration(train=3 secs)
Duration(test=whole) $lr$=$10^{-4}$
$m$=0.1 $s$=10 OneCycleLR

TABLE 3.8: Impact of OneCycleLR scheduler base *lr*

Table 3.8 shows the results of testing several base *lr* values for OneCycleLR scheduler. The one we were using (0.1) gave the best results among the three values tested by about 2 points for *UAR* and *Acc*. Those results can be explained because having a higher base *lr* allows the model to have a wider range to learn better by adjusting the *lr* at each batch (for OneCycleLR). Moreover, the fact that there is only once cycle helps the model learning more precisely to recognize features associated to each emotion. To conclude about this section, we kept OneCycleLR with a base *lr* at 0.1 for the next experiments.

## 3.3 Impact of the batch size

In this section, we wanted to know which batch size gives the best performance. We were using a batch size of 200, with 50 examples per class, and wanted to see the impact of a different sizes on the results.

| Ref | Batch size | *UAR*(%) | *Acc*(%) |
|---|---|---|---|
| a | 80 | 70.3 ±1.0 | 69.2 ±1.0 |
| b | 200 | **73.4** ±1.0 | **72.0** ±1.0 |
| c | 300 | 72.0 ±1.0 | 70.3 ±1.0 |
| d | 400 | 69.4 ±1.0 | 69.1 ±1.0 |

Duration(train=3 secs) Duration(test=whole) $lr$=$10^{-4}$
$m$=0.1 $s$=10 OneCycleLR(base $lr$=0.1)

TABLE 3.9: Impact of the batch size

Table 3.9 presents the results: we can see model (b) gave the most performant model while model (d) performed the worst. A batch size of 400 equals to 100 examples per class, which seems good. If we have more examples, the model should be able to learn the differences between each class, and be more precise to recognize them. However, according to the results, it is not the case. By giving many examples, the model get used to those ones, and does not adapt to unseen examples. Model (d) result is surprising, but can be questioned with our number of epochs (15) and the $lr$. Further experiments are necessary, but for computational cost and time lacking we have decided to not pursue them.

It shows how important it is to chose a correct batch size: if it is too low, the model will focus on the noise from the training data (model (a)), but if it is too big, the model will may not be able to generalize (model (d)).

After keeping model (b), we started to examine the possibility of taking only a few seconds of each speech turns for both the training and the testing processes.

## 3.4 Impact of speeches duration for training and testing

### 3.4.1 Impact of the training and testing durations

In this section, we wanted to explore the possibility of using either a few seconds of each speech segment, either the full speech segment thanks to SIDEKIT training duration parameter. This parameter can be changed for both training and validation sets, but we kept the same value for both of them for all our experiments.

Training duration of 3 means we keep speech turns with a length above 3 seconds (those below or equal are not used), and we select randomly 3 seconds in those speech turns. Training duration -13 means we select the speech with a length equal or lower than 13 seconds without cutting it. The speech turns are padded to have a duration of 13 seconds if needed. The test set is the same whatever was the training duration, and the whole test speech turns are used no matter their duration.

The initial thought was using the entire speech turn would give better results since there are more information. However, Schuller and Devillers, 2010 showed 1 second of speech is sufficient to recognize the emotion. We tested both with our custom model and data. We chose a batch size of 80 because of the computational cost of the model with a training duration of -13. Even if the training set was the same for all the training duration tested, not all speech turns were used for the training due to their duration.

| **Training duration** | ***UAR*(%)** | ***Acc*(%)** |
|---|---|---|
| -13 | 52.3 ±1.0 | 48.7 ±1.0 |
| 3 | **70.3** ±1.0 | **69.2** ±1.0 |
| 1 | 66.4 ±1.0 | 65.0 ±1.0 |

Batch size=80 $lr$=$10^{-4}$ Duration(test=whole)
$m$=0.1 $s$=10 OneCycleLR(base $lr$=0.1)

TABLE 3.10: Impact of speech turns training duration

Table 3.10 displays the results of mode trained on several training durations. They shows our initial thought was wrong: the best results are achieved by a training duration of 3. Keeping all the speech gave the worst results, which can be explained by the fact that only a part of speech turns contains the emotional information. If we keep the entire speech turn, we also keep the noise and consequently, prevent our model from efficiently learning each emotion features. The reverse thought can be applied to a training duration of 1: by keeping only 1 second of speech turn, we do not get enough information to understand each emotion features, and so to classify new speech turns.

After seeing the impact of the training duration, we wondered if applying the same idea to the testing duration would impact the results, i.e. if keeping the whole speech gave better results than taking only 1 or 3 seconds, and if the position of those few seconds mattered or not. For those and future experiments, we came back to a batch size of 200.

| Testing duration | Position | *UAR*(%) | *Acc*(%) |
|---|---|---|---|
| 3 | Beginning | 72.4 ±1.0 | 71.1 ±1.0 |
| | Random | 72.3 ±1.0 | 71.0 ±1.0 |
| All speech | - | **73.4** ±1.0 | **72.0** ±1.0 |

Batch size=200 Duration(train=3 secs)
$lr$=$10^{-4}$ $m$=0.1 $s$=10 OneCycleLR(base $lr$=0.1)

TABLE 3.11: Impact of speech turns test duration

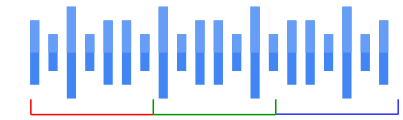
Table 3.11 displays the results of selecting either 3 seconds (and where) or the whole speech for testing duration. "Beginning" position means the 3 seconds were taken from the first 3 seconds of the speech segment while "Random" means we randomly extracted 3 seconds in the whole speech segment. Contrary to the training duration, keeping the whole speech segment for testing duration performed better than keeping only 3 seconds of it, no matter its position. An explanation can be that the model is seeking for specific features in the whole speech turn, and if we keep only 3 seconds of the testing speech turns, the model will not be able to find the features easily.

From now on, we will continue to use a training duration of 3, and the whole speech turn for the testing part.

### 3.4.2 Impact of SIDEKIT overlap parameter

An other important parameter within SIDEKIT is the overlap. Indeed, SIDEKIT is slicing each speech turn with a duration above the training duration parameter in chunks of a duration equals to the training duration. The overlap parameter is a number randomly set for each chunk: it can be equal to 0 (nothing changes) or it can be positive or negative (the chunk will be shifted). Figure 3.3 illustrates the three steps.

- First, the speech turn is splitted into different chunks
- Secondly, the overlap value is randomly set
- Finally, the start of the chunk is shifted according to the overlap value (its duration remains the same)



Chunk #1 Chunk #2 Chunk #3

(A) First step

Overlap=1 Overlap=0 Overlap=-1

(B) Second step

Chunk #1 Chunk #3

Chunk #2

(C) Last step

FIGURE 3.3: Overlap parameter

In order to test different values of overlap, we tested in parallel several training duration values from 1 to 5 seconds. There are two tests for each training duration: the first one is with an overlap equal to the training duration, and the second is either equal to 2 (for the training duration equal to 1), either equal to 3. Therefore, the three overlap options (equal, lower or bigger than the training duration) were tested.
For the experiment where the training duration is equal to 1, we were not able to use an overlap of 3 due to computational issues. This is the reason why we tested an overlap of 2 and not of 3 like the other experiments.

| Training duration | Overlap | *UAR*(%) | *Acc*(%) |
|---|---|---|---|
| 1 | 1 | 70.3 ±1.0 | 69.6 ±1.0 |
| | 2 | 71.8 ±1.0 | 71.2 ±1.0 |
| 2 | 2 | **74.9** ±1.0 | **74.0** ±1.0 |
| | 3 | 72.6 ±1.0 | 71.2 ±1.0 |
| 3 | 3 | 73.4 ±1.0 | 72.0 ±1.0 |
| 4 | 3 | 69.2 ±1.0 | 67.7 ±1.0 |
| | 4 | 68.4 ±1.0 | 67.0 ±1.0 |
| 5 | 3 | 67.4 ±1.0 | 65.5 ±1.0 |
| | 5 | 68.3 ±1.0 | 66.9 ±1.0 |

Batch size=200 Duration(test=whole) $lr$=$10^{-4}$
$m$=0.1 $s$=10 OneCycleLR(base $lr$=0.1)

TABLE 3.12: Impact of the overlap and the training duration

Table 3.12 summarizes the results for several training duration and their associated overlap values. The best result was achieved by a training duration of 2 with an overlap of 2.

Lets analyze first the case where the overlap is equal to the training duration, in second when it is smaller and finally when it is bigger.

We can see when the overlap is equal to the training duration that strictly above 3 seconds, the performance are decreasing drastically (around 4 points less for *UAR* and 5 points less for *Acc* than 3 seconds results). Keeping too many speech segments degrades emotional recognition because of the noise around emotional features. As seen earlier, a training duration of 1 performed worst than a training duration of 3. However, a training duration of 2 performed better than a training duration of 3 with 1.5 points difference, which can be surprising. Those results means emotional features can be contained in only 2 seconds of speech with as little as possible noise inside those seconds.

For the case where the overlap is smaller than the training duration (only for training duration of 4 and 5 seconds), we can see an improvement of around 2 points for an overlap of 3. This means some segments inside speech prevent emotion from being recognized.

Finally, when the overlap is bigger than the training duration (only for training duration of 1 and 2), we can see a training duration of 2 with an overlap of 3 performed better than a training duration of 1 with an overlap of 2 (around 2 points). Nevertheless, those results have to be taken carefully since the overlap tested were different.

To conclude this part about training duration, testing duration and overlap, we achieved the best results with a training duration of 2 seconds, an overlap equals to the training duration, and by using the whole speech for testing.

After testing the impact of different parameters (the loss hyperparameters, the scheduler, the batch size, the duration), we asked ourselves if we could still improve our model performance. After a review of our model parameters, we realized we forgot to test one of the main parameter: the learning rate *lr*.

## 3.5 Impact of the learning rate

Before beginning our experiments, we have decided to set the *lr* to $10^{-4}$ and to not change it since it was the *lr* value used in Chen et al., 2021, which has the best results in emotion recognition (70.6% of *Acc*). They used both a *lr* of $10^{-5}$ (for WavLM Large) and $10^{-4}$ (for both WavLM Base and WavLM Base+) for their model, but we have decided to use the second value, the first one seeming too low. However, since they pre-trained their model, Chen et al., 2021 used a *lr* of $10^{-3}$ for WavLM Large.

We then decided to test different *lr* to see the impact on our model. Since we kept OneCycleLR as our scheduler, it means the minimum *lr* before the decay phase is either $10^{-3}$, $10^{-4}$, or $10^{-6}$, and the maximum *lr* is 0.1.

| LR | ***UAR*(%)** | ***Acc*(%)** |
|---|---|---|
| $10^{-3}$ | **75.3** ±1.0 | **74.0** ±1.0 |
| $10^{-4}$ | 74.9 ±1.0 | 74.0 ±1.0 |
| $10^{-6}$ | 74.7 ±1.0 | 73.4 ±1.0 |

Batch size=200
Duration(train and overlap=2)
Duration(test=whole)
*m*=0.1 *s*=10
OneCycleLR(base *lr*=0.1)

TABLE 3.13: Impact of learning rate

Table 3.13 displays the results of the different tested *lr*. A *lr* of $10^{-3}$ performed better than both a *lr* of $10^{-4}$ and a *lr* of $10^{-6}$ by around 1 point for *UAR*: a bigger *lr* helps the model to better recognize emotions.

## 3.6 Conclusion of the experiments

To conclude our experiments, our best model have these hyperparameters and parameters values:

- A batch size of 200
- A training duration and an overlap of 2
- A margin $m$ of 0.1 and a feature scale $s$ of 10 for the Additive Angular Margin (*aam*) loss
- The OneCycleLR scheduler with a base $lr$ of 0.1
- A $lr$ of $10^{-3}$

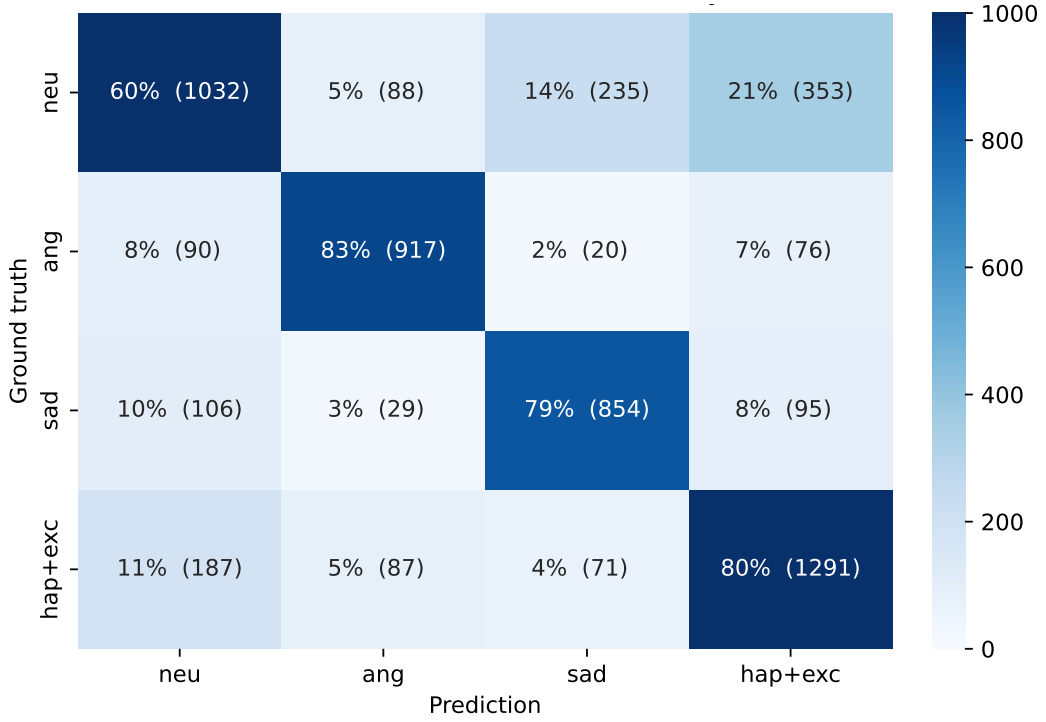



FIGURE 3.4: Confusion matrix of our best model

Figure 3.4 is the confusion matrix of our more performant model. We can see *neu* is the weakest point, with only 60% of *Acc* while the other classes achieved at least 79% of *Acc*. Our model is capable of detecting the correct emotion three times out of four.

For comparison, Nourtel et al., 2022 used a Support Vector Machine (*SVM*) with the same dataset and achieved 57.1% *UAR*, which is outperformed by our model. Moreover, our results are significant for the state of the art since we outperformed the best models for both *UAR* and *Acc*. Indeed, our result is greater than the state of the art confidence interval (66.0 ± 1.0% for *UAR* and 70.6% ± 1.0% for *Acc*).

## 3.7 Future work

Three future works are interesting.

The first one is the continuation of the ANR project Deep-Privacy. This internship work aimed at building an efficient Speech Emotion Recognition model on original (non-anonymized) speech signals, which we achieved. Our model is able to recognize a speech signal emotion three times out of four. The next step is to train and test this model on anonymized speech. According to the obtained results, and if emotions are considered as sensitive information or not, we will either try to modify the conversion process of anonymization (according to the final goal) or either improve the emotion recognition with anonymized speech.

The second is the adaptation of the model to other datasets and other emotions. Is our model able to recognize the 4 emotions from other datasets? Will the recognition be also performant if we add several other emotions? Our model was tested on unseen data, but since it was the same dataset, the recording parameters were the same. The question is to know if the model will recognize emotion with different recording parameters, or even with speech recorded with smartphones. To summarise, the robustness of our model has to be experienced with dissimilar data and recording conditions.

Finally, the last is a deeper analyzis of *IEMOCAP*. Indeed, during our experiments, we suspected that some *IEMOCAP* speech turns prevented the model from learning emotional features because of their noise, or their annotation. We listened to some speech turns and find their annotations questionable. It is the eternal issue with subjective perception, but we lacked time to deepen this analyzis.

Both emotion and speaker recognition are growing and vital domains in our era. More and more progress will be made within the next few years, for our privacy sake.

# Chapter 4

# Conclusion

## 4.1 Internship

Doing research for five months inside the MULTISPEECH team was interesting, both professionally and personally.

I had the opportunity to work with several people, from interns like me to researchers and teacher-researchers, inside one of the biggest teams within Inria. There was a wide diversity in the domains studied (hate speech detection, talking heads...) and despite this variety, everyone was available if I had a question. The seminars were really interesting and happened quite a few times about different topics. I learned many things from them. Moreover, I had the chance to be present in the facility every day to work, which is great for both work and socialization.

Concerning my work, my supervisors were always present if I had any questions or ideas. The weekly meetings helped me to stay focused and to prepare my weekly advances, questions, and issues. The first weeks were dedicated to the bibliography, reading the state of the art for emotion recognition, but also from other domains such as speaker recognition. After getting a large amount of information, I started to get acquainted with SIDEKIT package, which I never heard about before this internship. A member of the team, Pierre CHAMPION, helped me with my inquiries every time, and even beyond those. He was present in the weekly meetings and was very helpful in responding to my questions and came up with ideas that helped throughout my internship. I truly felt I was not alone for those entire five months, which helped me to work. My first internship in this team (from July to early September 2021 in the same field) was also interesting, but I just barely touched the work of a researcher. Now, thanks to this 2022 internship, I understand better the research world and can say I want to be part of it.

Regarding the personal benefits of this internship, I feel more confident in my abilities. Since the beginning of this Master, I have the impostor syndrome. It is the feeling that you are a fake, and everything you do does not deserve any recognition since you are not worth it. Other people make wonderful work and, even if you do the same as them, your work will always fall short of their work. Sometimes, you know you are doing good, but a minor error is sufficient to think negatively about yourself for several days. This mental state yo-yo happened to me during the internship, and even if I did not talk about it to my supervisors (because I feel this is my burden), they did nothing to lower me down. Every word, question, and talk was not negative at all, even when I did mistakes. They were simply understanding and gave me the advice to help me not reproduce those. I knew every day that, despite the fact I was feeling down, there would be a ray of sunshine in the office from my

coworkers to help me lighten my mind. This internship helped me understand better my value, strengths, and my weaknesses (to overcome them).

To conclude, my internship was the best I could even imagine, with my supervisors and the MULTISPEECH team. I will treasure all the memories from it for my life.

## 4.2 Master Degree

In this section, I want to focus specifically on the Master. Before being accepted into it, I would never have thought I will do a Master (and even a Ph.D.). I was not sure if I would be able to achieve it, since I never studied programming before. My linguistic background felt insufficient compared to the whole computational science I was about to dive into.
The first weeks were complicated, for several reasons. First, it was the first time I was alone in a foreign region, far away from my family. Second, it was the first time I will have to speak every day in English. Finally, the gap between my background and what I thought I had to know was huge and scary. I went to the first classes with all those fears.

The first year was complete and difficult. The teachers were patient with us, knowing we had different backgrounds and tried their best to teach us despite this difficulty. The student teamwork was also very beneficial for everyone and personally helped me understand better some courses. Some difficulties happened, like the lockdown for several months. Those were hard times, both for students and teachers. Hopefully, the second year did not begin in lockdown.

The first semester of the second year was intense. There were many courses, and sometimes several parts which could have been their own separate course. We felt under pressure every day and tried to decompress when we could. Moreover, the searches for an internship and the administrative links to it added more pressure over time. Like the first year, the teachers gave their best to teach us and were adaptable for projects due considering the amount of work we had to do.

To conclude, this Master was intense, difficult, and very interesting. Yes, we had hard times, but in the end, I felt we have all the keys to work in the NLP domain, whether in companies or research.

# Appendix A

# Information about emotions

| | |
|---|---|
| 1. | The concept of emotion is applicable to all evolutionary levels and applies to all animals including 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. |

TABLE A.1: The ten postulates of Plutchik's theory of basic emotions

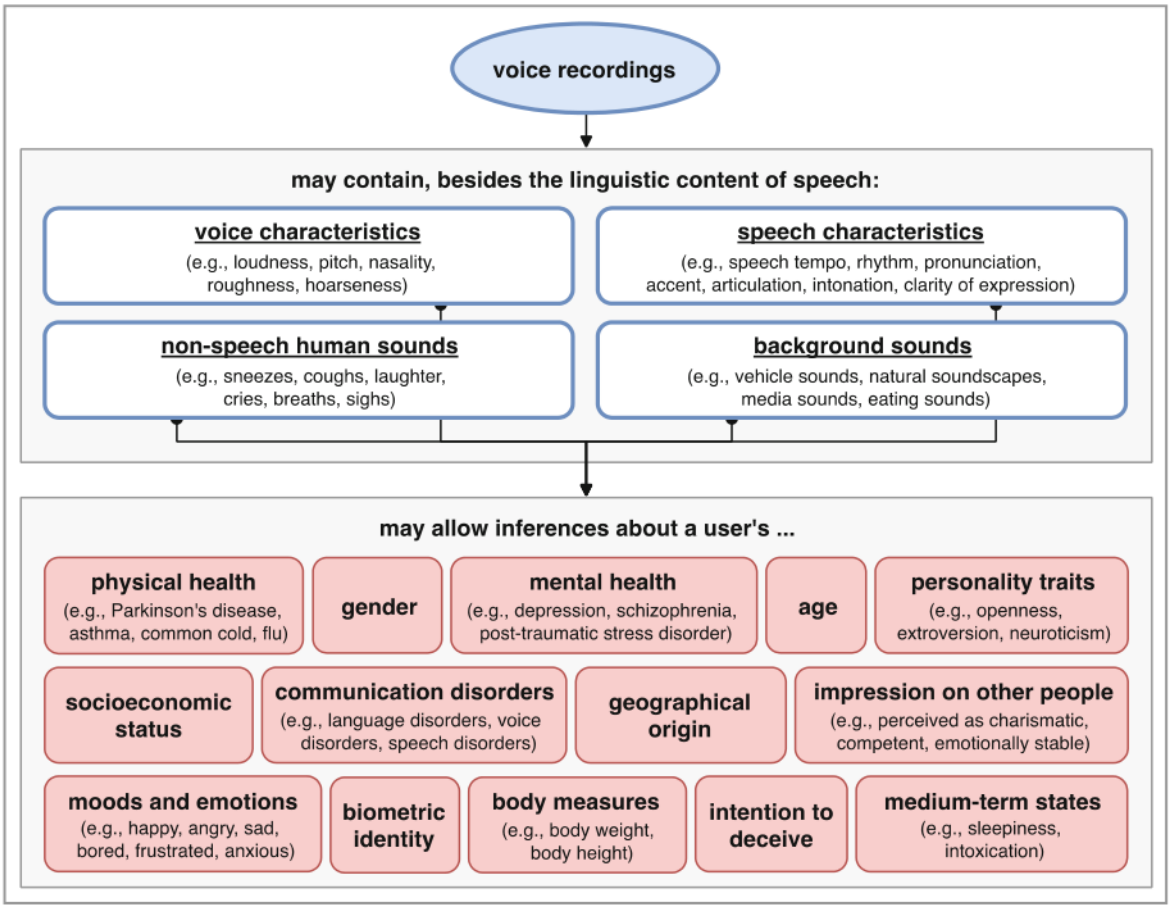



FIGURE A.1: Sensitive information discernable from speech (Kröger, Lutz, and Raschke, 2020)

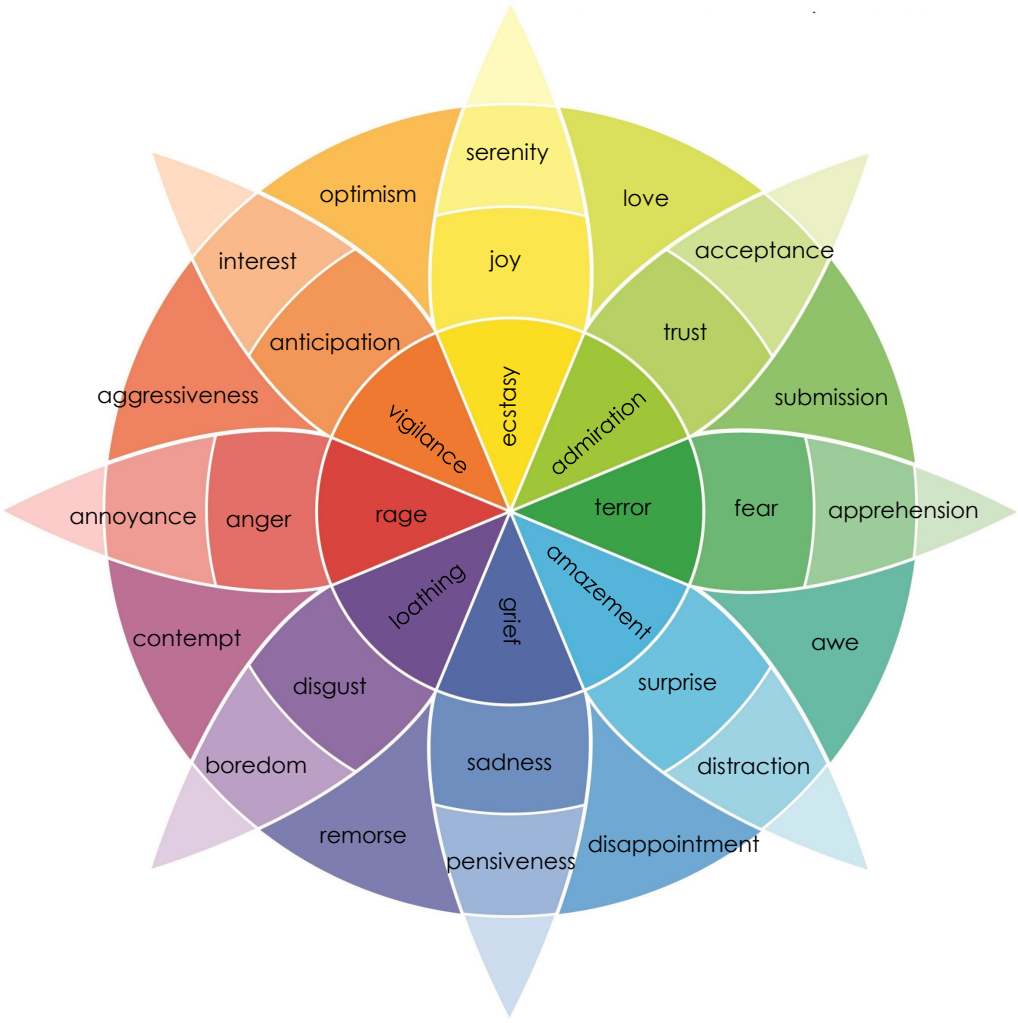



FIGURE A.2: Plutchik's Wheel of Emotions

# Appendix B

# Information about IEMOCAP

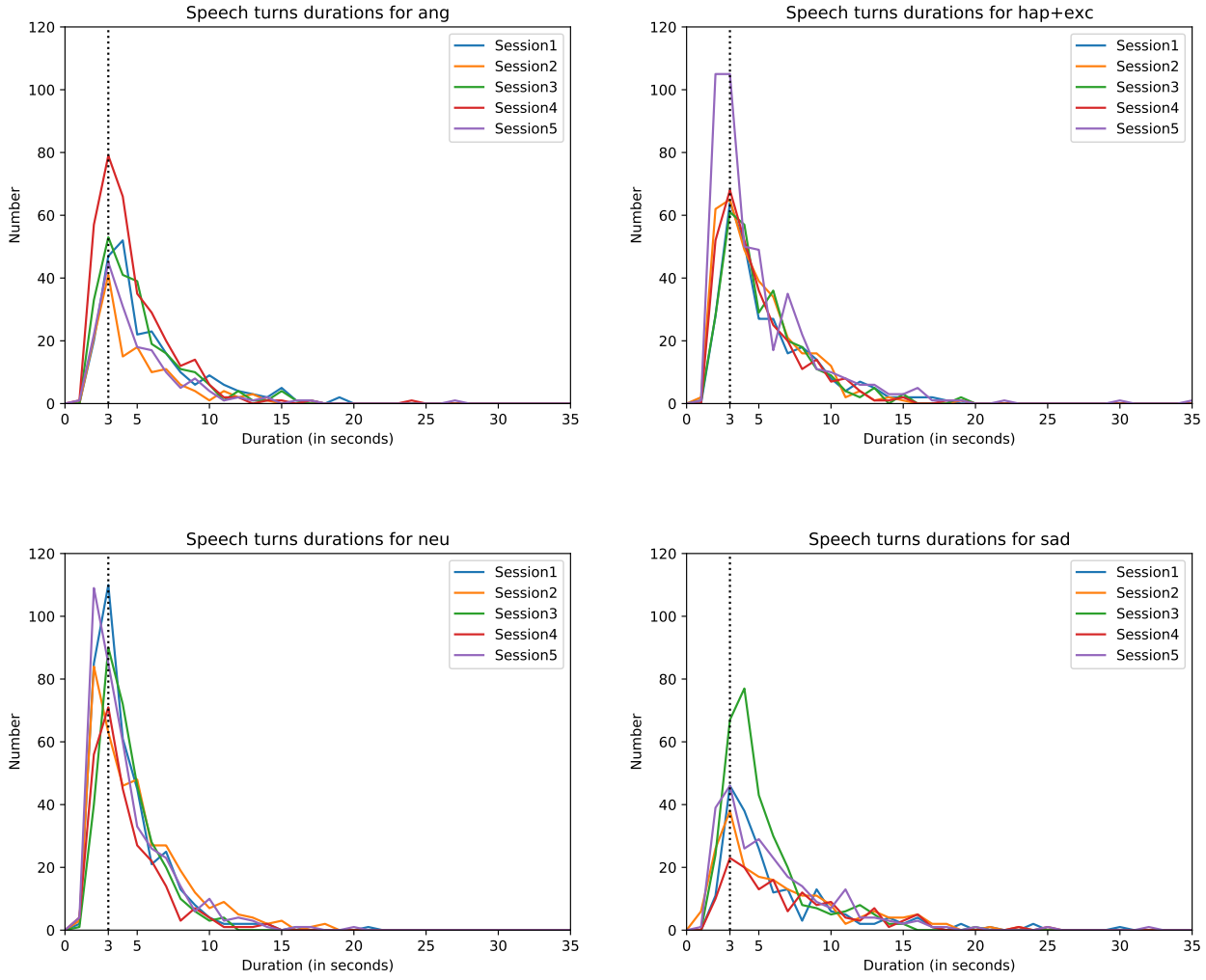



FIGURE B.1: Distribution of the duration of speech turns by session of the considered emotions in *IEMOCAP*

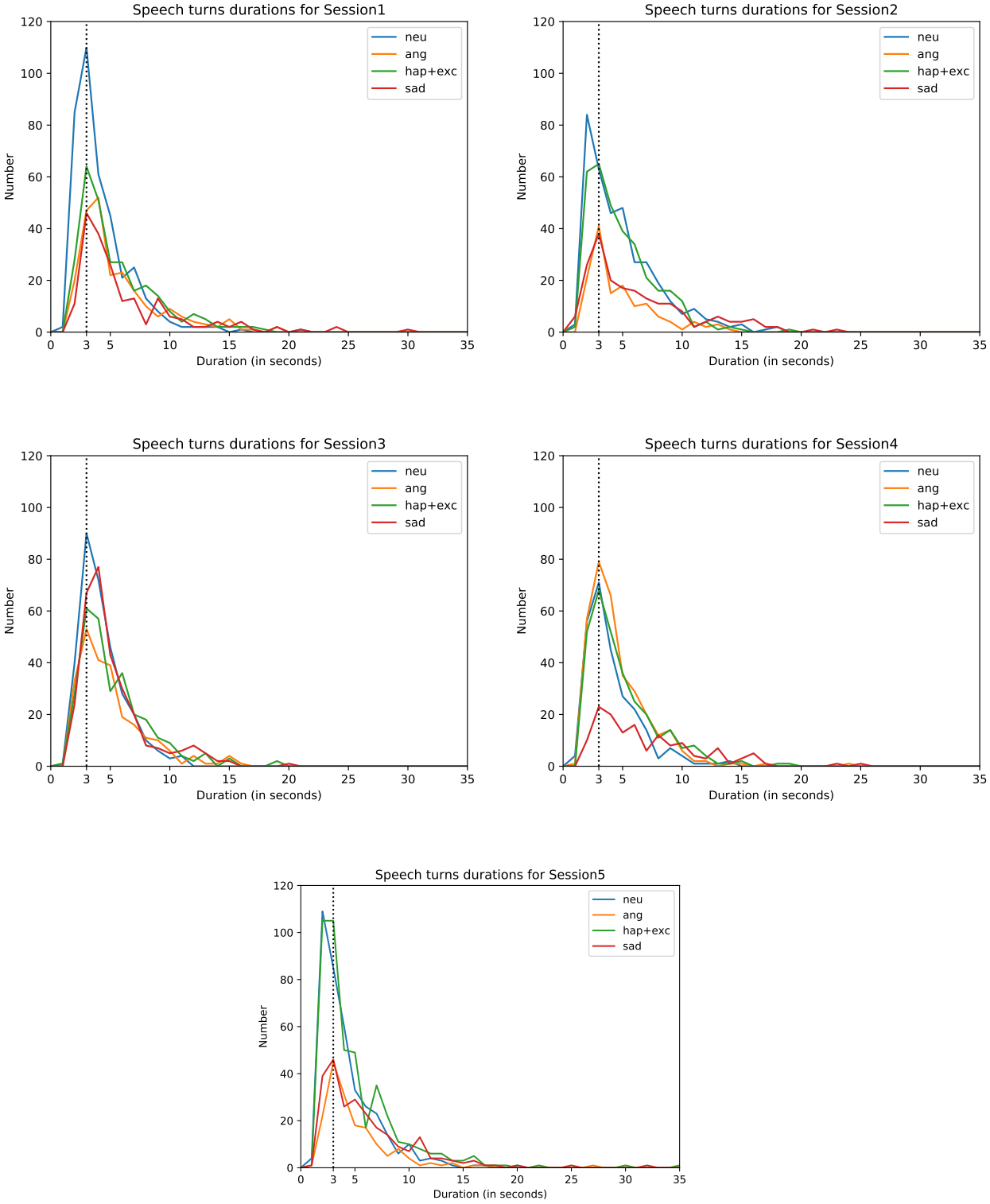
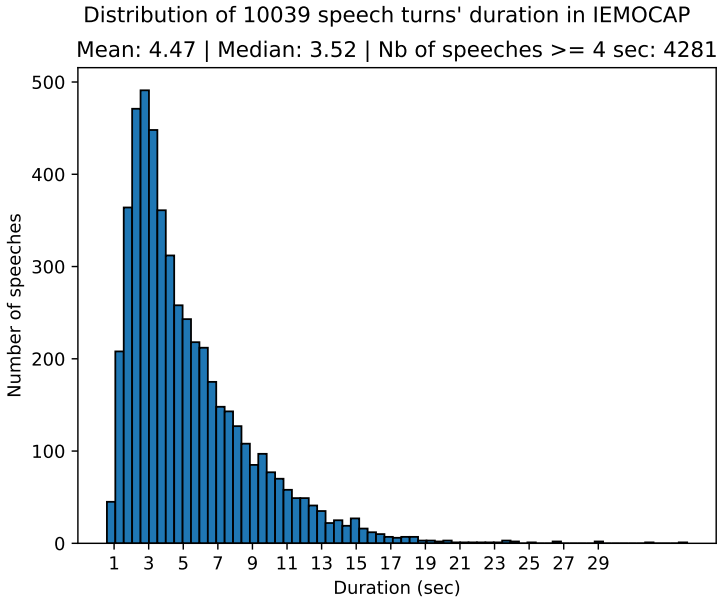
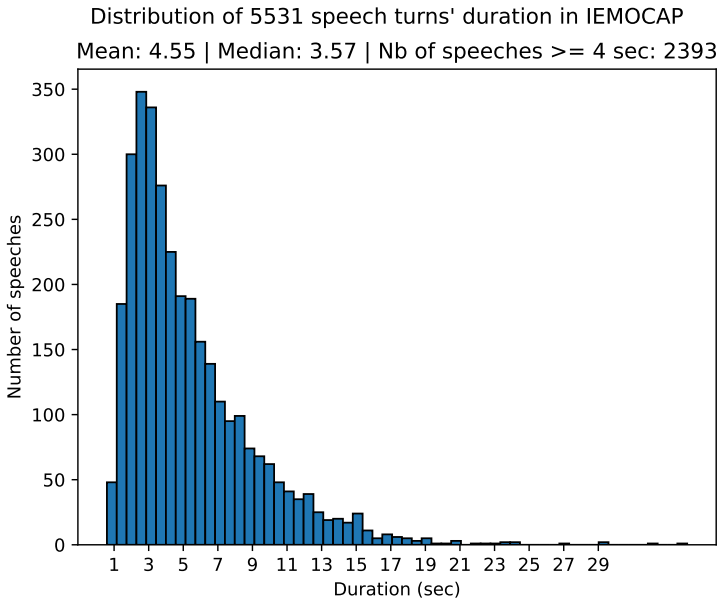



FIGURE B.2: Distribution of the duration of speech turns by emotion for each session in *IEMOCAP*




(A) 10039 speech turns (all emotions considered)




(B) 5531 speech turns (*neu*, *ang*, *sad*, *hap+exc* considered)

FIGURE B.3: Distribution of duration of speech turns in *IEMOCAP*

# Appendix C

# Information about annotations

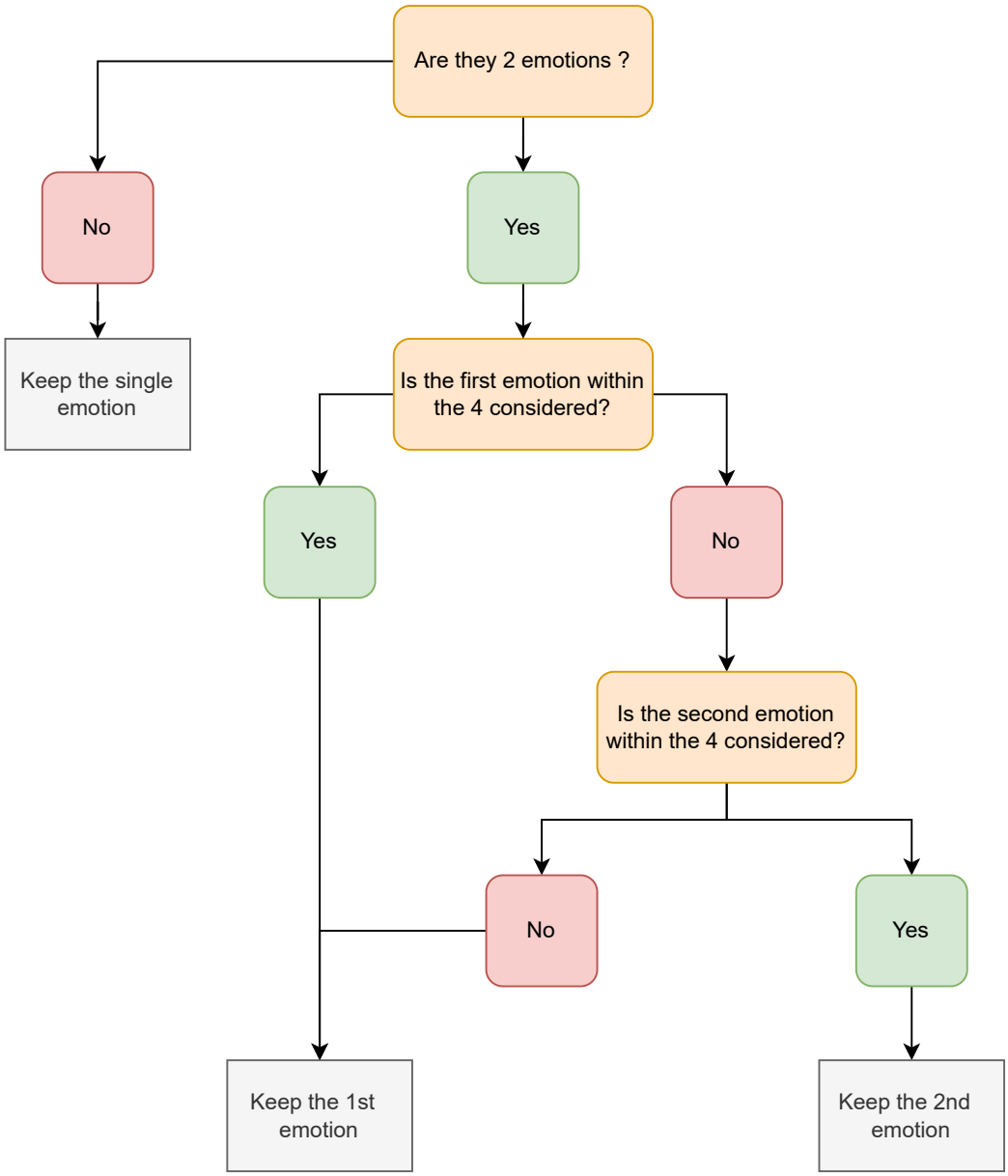



FIGURE C.1: Guideline to know which annotation to chose for each annotator

# Bibliography

Busso, Carlos, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan (2008). "IEMOCAP: interactive emotional dyadic motion capture database". In: *Language Resources and Evaluation*, pp. 335–359.

Cai, Xingyu, Jiahong Yuan, Renjie Zheng, Liang Huang, and Kenneth Church (2021). "Speech Emotion Recognition with Multi-Task Learning". In: *Interspeech 2021*. International Speech Communication Association (ISCA), pp. 4508–4512.

Chen, Sanyuan, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, and Furu Wei (2021). "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". In: *Computing Research Repository (CoRR)*.

Contributors to Wikipedia projects (n.d.). *Robert Plutchik*.

Deng, Jiankang, Jia Guo, Niannan Xue, and Stefanos Zafeiriou (2019). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers (IEEE), pp. 4685–4694.

Desplanques, Brecht, Jenthe Thienpondt, and Kris Demuynck (2020). "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification". In: *Interspeech 2020*. International Speech Communication Association (ISCA), pp. 3830–3834.

Ekman, Paul (1999). "Basic Emotions". In: *Handbook of Cognition and Emotion*. John Wiley  Sons, Ltd. Chap. 3, pp. 45–60.

Eyben, Florian, Klaus R. Scherer, Bjorn W. Schuller, Johan Sundberg, Elisabeth Andre, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong (2016). "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing". In: *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Affective Computing*, pp. 190–202.

Fahad, Md Shah, Ashish Ranjan, Akshay Deepak, and Gayadhar Pradhan (2022). "Speaker Adversarial Neural Network (SANN) for Speaker-independent Speech Emotion Recognition". In: *Circuits, Systems, and Signal Processing*.

Fang, Fuming, Xin Wang, Junichi Yamagishi, Isao Echizen, Massimiliano Todisco, Nicholas Evans, and Jean-Francois Bonastre (2019). "Speaker Anonymization Using X-vector and Neural Waveform Models". In: *10th ISCA Workshop on Speech Synthesis (SSW 10)*. International Speech Communication Association (ISCA), pp. 155–160.

Fayet, Cédric, Arnaud Delhay, Damien Lolive, and Pierre-François Marteau (2018). "EMO&LY (EMOtion and AnomaLY) : A new corpus for anomaly detection in an audiovisual stream with emotional context". In: *Language Resources and Evaluation Conference (LREC)*.

Fér, Radek, Pavel Matějka, František Grézl, Oldřich Plchot, Karel Veselý, and Jan Honza Černocký (2017). "Multilingually trained bottleneck features in spoken language recognition". In: *Computer Speech & Language*, pp. 252–267.

Gorrostieta, Cristina, Reza Lotfian, Kye Taylor, Richard Brutti, and John Kane (2019). "Gender De-Biasing in Speech Emotion Recognition". In: *Interspeech 2019*. International Speech Communication Association (ISCA), pp. 2823–2827.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Kröger, Jacob Leon, Otto Hans-Martin Lutz, and Philip Raschke (2020). "Privacy Implications of Voice and Speech Analysis – Information Disclosure by Inference". In: *Privacy and Identity Management. Data for Better Living: AI and Privacy*. Ed. by Michael Friedewald, Melek Önen, Eva Lievens, Stephan Krenn, and Samuel Fricker. Springer International Publishing, pp. 242–258.

Larcher, Anthony, Kong Aik Lee, and Sylvain Meignier (2016). "An extensible speaker identification sidekit in Python". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5095–5099.

Lee, Chi-Chun, Emily Mower, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan (2011). "Emotion recognition using a hierarchical binary decision tree approach". In: *Speech Communication*, pp. 1162–1171.

Li, Yuanchao, Tianyu Zhao, and Tatsuya Kawahara (2019). "Improved End-to-End Speech Emotion Recognition Using Self Attention Mechanism and Multitask Learning". In: *Interspeech 2019*. International Speech Communication Association (ISCA), pp. 2803–2807.

Lotfian, Reza and Carlos Busso (2019). "Building Naturalistic Emotionally Balanced Speech Corpus by Retrieving Emotional Speech from Existing Podcast Recordings". In: *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Affective Computing*, pp. 471–483.

Macary, Manon, Marie Tahon, Yannick Esteve, and Anthony Rousseau (2021). "On the Use of Self-Supervised Pre-Trained Acoustic and Linguistic Features for Continuous Speech Emotion Recognition". In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. Institute of Electrical and Electronics Engineers (IEEE), pp. 373–380.

Nourtel, Hubert, Pierre Champion, Denis Jouvet, Anthony Larcher, and Marie Tahon (2021). "Evaluation of Speaker Anonymization on Emotional Speech". In: *2021 ISCA Symposium on Security and Privacy in Speech Communication*. International Speech Communication Association (ISCA), pp. 62–66.

— (2022). "Analyse de l'anonymisation du locuteur sur de la parole émotionnelle". In: *JEP 2022 (Journées d'Études sur la Parole)*.

Pappagari, Raghavendra, Tianzi Wang, Jesus Villalba, Nanxin Chen, and Najim Dehak (2020). "X-Vectors Meet Emotions: A Study On Dependencies Between Emotion and Speaker Recognition". In: *ICASSP 2020 - IEEE International Conference on Acoustics, Speech and Signal Processing (International Conference on Acoustics, Speech and Signal Processing (ICASSP))*. Institute of Electrical and Electronics Engineers (IEEE), pp. 7169–7173.

Schmitt, Maximilian, Nicholas Cummins, and Björn W. Schuller (2019). "Continuous Emotion Recognition in Speech — Do We Need Recurrence?" In: *Interspeech 2019*. International Speech Communication Association (ISCA), pp. 2808–2812.

Schuller, Björn and Laurence Devillers (2010). "Incremental acoustic valence recognition: An inter-corpus perspective on features, matching, and performance in a gating paradigm". In: pp. 801–804.

Smith, Leslie N. (2017). "Cyclical Learning Rates for Training Neural Networks". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Institute of Electrical and Electronics Engineers (IEEE), pp. 464–472.

Smith, Leslie N. and Nicholay Topin (2019). "Super-convergence: very fast training of neural networks using large learning rates". In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Ed. by Tien Pham. Society of Photographic Instrumentation Engineers (SPIE), p. 36.

Tahon, Marie and Laurence Devillers (2016). "Towards a Small Set of Robust Acoustic Features for Emotion Recognition: Challenges". In: *Institute of Electrical and Electronics Engineers (IEEE)/ACM Transactions on Audio, Speech, and Language Processing*, pp. 16–28.

Trigeorgis, George, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A. Nicolaou, Bjorn Schuller, and Stefanos Zafeiriou (2016). "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (International Conference on Acoustics, Speech and Signal Processing)*. Institute of Electrical and Electronics Engineers (IEEE), pp. 5200–5204.

Tsai, Hsiang-Sheng, Heng-Jui Chang, Wen-Chin Huang, Zili Huang, Kushal Lakhotia, Shu-wen Yang, Shuyan Dong, Andy Liu, Cheng-I Lai, Jiatong Shi, Xuankai Chang, Phil Hall, Hsuan-Jui Chen, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee (2022). "SUPERB-SG: Enhanced Speech processing Universal PERformance Benchmark for Semantic and Generative Capabilities". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 8479–8492.

Wang, Feng, Jian Cheng, Weiyang Liu, and Haijun Liu (2018). "Additive Margin Softmax for Face Verification". In: *Institute of Electrical and Electronics Engineers (IEEE) Signal Processing Letters* 25, pp. 926–930.

Wang, Feng, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille (2017). "NormFace: L $_2$ Hypersphere Embedding for Face Verification". In: *Proceedings of the 25th ACM international conference on Multimedia*. Mountain View California USA: Association for Computing Machinery (ACM), pp. 1041–1049.

Yeh, Sung-Lin, Yun-Shao Lin, and Chi-Chun Lee (2020). "Speech Representation Learning for Emotion Recognition Using End-to-End ASR with Factorized Adaptation". In: *Interspeech 2020*. International Speech Communication Association (ISCA), pp. 536–540.