Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Bachelorarbeit (Informatik)

Speech Classification using wav2vec 2.0

Autoren	Pascal Fivian Dominique Reiser
Hauptbetreuung	Prof. Dr. Mark Cieliebak

Datum	11.06.2021

Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

Erklärung betreffend das selbständige Verfassen einer Bachelorarbeit an der School of Engineering

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

Zürich, 11.6.2021

Unterschriften:

Pascal Fivian

Dominique Reiser

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Bachelorarbeiten zu Beginn der Dokumentation nach dem Titelblatt mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

Abstract

The Wav2Vec 2.0-XLSR-53 is a powerful model that was pre-trained to learn multilingual speech representation end-to-end in an unsupervised way. Dialect Identification (DID) and Accent Identification (AID) can be used to improve Automatic Speech Recognition (ASR) systems in languages with multiple distinctive dialects or accents. This thesis uses a classifier on top of wav2vec to classify speech. It is evaluated how the model performs when trained on low-resource datasets. Various experiments are conducted in the areas of AID in English and Spanish. In addition, evaluations were executed on short samples. To further explore the capabilities of wav2vec, an age and sex classifier is trained on German speech. The used corpora were extracted from Mozilla's Common Voice (Common Voice). Trained was on 1.5 up to 8 hours per class. An average F1-score of 0.396 could be achieved for English while classifying six accents. On seven Spanish accents, an F1-score of 0.266 was reached. In the area of sex identification, an F1-score of 0.90 was reached while achieving 0.360 for age identification with a Macro Averaged Mean Absolute Error (MAEM) of 0.982. The results show that it is possible to train a classifier on wav2vec. However, the achieved scores do not correspond to the desired values. It is shown that for training a better classifier, longer and more samples are needed. Furthermore, it is important for the collection of training-data that the samples contain recordings with freely-spoken speech that is not read from a text.

Zusammenfassung

Wav2Vec 2.0-XLSR-53 ist ein leistungsfähiges Modell, das pre-trained wurde, um mehrsprachige Sprachrepräsentationen end-to-end zu erlernen. Dialekt-Identifikation (DID) und Akzent-Identifikation (AID) können verwendet werden, um automatische Spracherkennungssysteme (ASR) in Sprachen mit mehreren ausgeprägten Dialekten oder Akzenten zu verbessern. In dieser Arbeit wird ein Klassifikator auf Basis von wav2vec verwendet, um Sprache zu klassifizieren. Es wird evaluiert, wie das Modell performt, wenn es auf Datensätzen mit geringen Ressourcen trainiert wird. Es werden verschiedene Experimente in den Bereichen AID in Englisch und Spanisch durchgeführt. Zusätzlich wurden Evaluationen auf kurzen Samples durchgeführt. Um die Fähigkeiten von wav2vec weiter zu untersuchen, wird ein Alters- und Geschlechtsklassifikator in deutscher Sprache trainiert. Die verwendeten Korpora wurden aus Mozillas CommonVoice (Common Voice) extrahiert. Trainiert wurde auf 1.5 bis 8 Stunden pro Klasse. Bei der Klassifikation von sechs Akzenten konnte für Englisch ein durchschnittlicher F1-Score von 0.396 erreicht werden. Bei sieben spanischen Akzenten wurde ein F1-Score von 0.266 erreicht. Im Bereich der Geschlechtsidentifikation wurde ein F1-Score von 0,90 erreicht, während für die Altersidentifikation ein Wert von 0.360 mit einem Macro Averaged Mean Absolute Error (MAEM) von 0.982 erzielt wurde. Die Ergebnisse zeigen, dass es möglich ist, einen Klassifikator auf wav2vec zu trainieren, allerdings entsprechen die erreichten Scores nicht den gewünschten Werten. Es zeigt sich, dass für das Training eines besseren Klassifikators längere und mehr Audiodaten benötigt werden. Für die Sammlung von Trainingsdaten ist es wichtig, dass die Aufnahmen frei gesprochene Sprache enthalten, die nicht von einem Text abgelesen wurden.

Preface

While working on this thesis, we learned a lot. We came in contact with excellent frameworks and libraries like pytorch, huggingface.co, numpy, and wandb.ai. The opportunity to invest our time in something that is state-of-the-art in the field of speech recognition was challenging and very rewarding. We thank Prof. Dr. Mark Cieliebak and Jan Deriu for their countless ideas and inspiring discussions throughout the whole project.

Contents

1. Introduction

In the last decade, Automatic Speech Recognition (ASR) became an important research topic, as there are more and more use cases. For example, big companies provide voice assistants, smart home devices, meeting transcriptions or live subtitle generation. Methods like deep learning helped a lot address this technological hurdle, and therefore significant improvements were made, including the availability of massive amounts of data. One of the biggest trade-offs regarding the training of neural networks is the amount of data needed to achieve good results. Therefore those systems are mainly available in languages where that amount of data is available. Especially in Swiss German, the amount of freely available training data is minimal, and one has to rely on training methods that address the problem of training data volumes. These solutions are currently developing very quickly, and one of the most recent solutions is wav2vec 2.0 (wav2vec), which Facebook AI released in October last year. It is a framework that learns through pre-training, end-to-end language representations that can be fine-tuned on a specific language with only a small amount of training data. [1]

Since there are very different dialects in Swiss German, a future ASR system may have to be trained separately for different dialect groups to be more successful. For this purpose, this thesis examines whether wav2vec is also suitable for speech classification. There is already a paper that deals with wav2vec in language identification and proves that wav2vec extracts features that can be used for classification. [2] But, this paper worked with a wav2vec model that was only pre-trained on English data. In the meantime, however, a model was released that was trained on several languages simultaneously and thus should be more eligible for that case.

This new model, named XLSR for Cross-Language Speech Representation, is used in this thesis to perform language classification in different domains to give an outlook on what to look for when classifying later on Swiss dialects. For this purpose, experiments are conducted to identify a speaker's accent with little training data available. Furthermore, we analyse how a trained classifier behaves when it receives only very short inputs. The aim is to recognise a speaker's dialect as quickly as possible in an ASR system. In addition, we explore some other more unconventional classifications in further experiments, explicitly addressing the vast possibilities of transfer learning in speech processing using wav2vec.

1.1. Literature Review

Research in Dialect Identification (DID), and Accent Identification (AID) has been a big topic in recent years. As a result, many different approaches to address this challenge has been chosen.

In the dissertation of F. Baidsy [3], it is demonstrated that dialects can be distinguished on behalf of certain phones that are different in dialects. It also shows the possibility of improving ASR by identifying the corresponding dialect before the transcription.

In 2014, Lazaridis et al. attempted to identify Swiss-French regional accents based on Gaussian Mixture Modelling (GMM) with two different GMM-based algorithms. First, universal background modelling followed by maximum-a-posteriori adaptation and total variability (i-vector) modelling. The i-vector-based system outperformed the other by a relative improvement of 15.3 %. The best accuracy while classifying four regional accents was 38.5 %. [4]

In the INTERSPEECH 2016 Computational Paralinguistics Challenge, the subject was the identification of foreign English accents. [5] The winning system used an approach based on i-vectors, classifying 11 accents with an accuracy of 84 %. [6]

In [7], they classify Mandarin into 15 accents and explore bidirectional Long Short-Term Memory and i-vectors to model longer-term acoustic context. They reached an accuracy of up to 34.1 % with 15 accents to predict. They then grouped the accents into three groups based on their geographical features, which boosted their accuracy. They also showed that individual systems trained on these accents could yield Character Error Rate improvements with the classifier in front.

In the Arabic Speech Recognition in the Wild challenge 2017, one of the two tasks was identifying five Arabic dialects. The best participant was able to reach an average of 80 % accuracy using Generative Adversarial Networks. [8] Before the challenge in [9] a multi-class Support Vector Machine was used to differentiate between English and Arabic with an accuracy of 100 %. When, distinguishing between the five most common dialects, they achieved an accuracy of 59.2 %. In 2018 Suwon Shon et al. [10] combined an end-to-end and a Siamese neural network to classify the five Arabic dialects on the same dataset as in [8] and [9]. They achieved an accuracy of 78 %.

P. Praikh et al. in 2020 [11] introduced a fused system consisting of a Deep Neural Network, Recurrent Neural Network and a Convolutional Neural Network. They differentiate between three English accents, namely Spanish, American and Indian, with an accuracy of 68.7 %

A binary classifier between Indian and American English was trained in [12]. They used Mel-frequency cepstral coefficients feature extraction on a dataset with five speakers reaching an accuracy of 95 % achieved with a feed-forward neural network.

Fan et al. evaluated the capability of the pre-trained wav2vec for speaker verification and language identification. They added a fully connected layer on top of wav2vec's feature encoder to distinguish ten languages or 1'211 speakers. They run their experiments with a pre-trained wav2vec feature encoder, and a randomly initialised one. The results showed that the pre-trained feature encoder was able to retain distinguishable features for both tasks. However, the results for speaker verification were better than those for language identification. They further assumed the underlying issue being that the model was pre-trained on solely English and suggest pre-training on multiple languages could mitigate this issue. [2]

1.2. Outline

This thesis is divided into four parts. First, in the Foundation Chapter 2 the underlying concepts and tools needed for understanding are explained in detail. Then, in the experiments Chapter 3, the experimental setup and all conducted experiments are described. Third, the corresponding results are presented in Chapter 4. Finally, in Chapter 5, the results are discussed, and further research opportunities are proposed.

1.3. Terminology

wav2vec

Wav2vec is an ASR system designed by Facebook. There are multiple publications available, and they can be distinguished in the following four parts.

Definition 1.3.1 (**wav2vec 1.0**)**.** Wav2vec 1.0 is the first release and represents Facebook's attempt to learn latent speech representation described by A. Baevski et al. [13]

Definition 1.3.2 (**vq-wav2vec**)**.** Vq-wav2vec is the second release and extends wav2vec 1.0 with a quantization module described by A. Baevski et al. [14]

Definition 1.3.3 (**wav2vec 2.0**)**.** Wav2vec 2.0 is the third release described by A. Baevski et al. [1] and the first release including a Transformer module.

Definition 1.3.4 (**wav2vec 2.0 XLSR**)**.** Wav2vec 2.0 XLSR is the fourth release built on wav2vec 2.0 and the first one including multilingual pre-training on 53 languages described by A. Conneau et al. [15]

Dialect vs. Accent

In this thesis, the words dialect and accent are increasingly used. In order not to confuse the two, they are defined as follows:

Definition 1.3.5 (**Dialect**)**.** The Cambridge Dictionary defines a dialect as: "A form of a language that is spoken in a particular part of a country or by a particular group of people and that contains some words, grammar, or pronunciations (= the ways in which words are said) that are different from the forms used in other parts or by other groups."[16]

Definition 1.3.6 (**Accent**)**.** The Cambridge Dictionary defines an accent as: "The way in which people in a particular area, country, or social group pronounce words."[17]

This shows that an accent describes mainly the pronunciation, while a dialect contains much more. Therefore, the word dialect is only used when speakers in sound recordings are speaking freely. On the other hand, if the speakers are reading from a text, this is more in line with the definition of an accent. The individual speakers differ more on their pronunciation because they tend not to use words they usually do and build their sentences biased.

Sex vs. Gender

For some experiments in this thesis, the words sex and gender are used. To clarify in which context the words are used and why they have to be distinguished, the definition is given here:

Definition 1.3.7 (**Sex**)**.** The Office for National Statistics UK defines sex as: "referring to the biological aspects of an individual as determined by their anatomy, which is produced by their chromosomes, hormones and their interactions." [18]

Definition 1.3.8 (**Gender**)**.** The Office for National Statistics UK defines gender as: "a social construction relating to behaviours and attributes based on labels of masculinity and femininity; gender identity is a personal, internal perception of oneself and so the gender category someone identifies with may not match the sex they were assigned at birth." [18]

Since this thesis attempts to identify the sex of a speaker by voice alone, it is important to know under which term the relevant data was collected. For example, in the case of gender, people might identify as something other than their biological sex.

2. Foundations

2.1. Speech Processing

The history of Speech Processing goes back to the late 18th century, whereby the first attempts were made in producing speech rather than Speech Recognition. Resonance tubes connected to organ pipes were used to produce vowel-like sounds. [19] From trying to create a speaking machine, the focus shifted to recognising speech with technological advancements. In 1952 Bell Laboratories developed a system that was able to recognise isolated digits from one speaker. [20] Further development introduced a rule-based system where the speech recognition language was represented in a graph including grammatical rules or word orders. [19] [21] In the 1980s, statistical methods developed rapidly, and with the upcoming hidden Markov model [22], the foundation of the modern speech recognition system was laid out.

With the upcoming of neural networks, the possibility emerged of modelling complex patterns in speech data. Since 2014 end-to-end ASR systems got much interest in research as they reduced the training complexity. Connectionist Temporal Classification (CTC) based systems were introduced in 2014, these models were able to map acoustics to characters, but they rely on a language model to improve transcription quality. [23] Another attempt for end-to-end ASR are attention-based systems. They can learn all the components from the pronunciation- to the acoustic- and the language-model directly. [24]

2.2. wav2vec

The first generation of wav2vec 1.0 released in September 2019 was one of Facebook AI's initiatives [25] to improve speech recognition systems not only in terms of accuracy[1] but also in a massive reduction in needed training data and its corresponding training time. [13] Traditional systems required thousands of hours of transcribed training-data, which is always very hard to get, especially for the 7'000 languages [26] worldwide, from which most are rarely spoken, such as Swiss-German. Over time new research led to further improvements published in chronological order of vq-wav2vec [14], wav2vec 2.0 [1], and wav2vec XLSR [15].

Wav2vec 2.0, for the first time, tackles this problem by learning speech representation end-to-end from unlabeled data in pre-training. This approach of using unlabeled data to train is called unsupervised learning. The pre-trained model can then be used

[1] Accuracy in this context refers to multiple metrics like Word Error Rate, Character Error Rate or Phoneme Error Rate

by various speech recognition system to fine-tune them with labelled data on a particular task. [1] Furthermore, to understand how wav2vec works, the involved concepts are explained in this chapter.

2.2.1. Transformer

Until the proposed concept of Transformers by A. Vaswani et al. [27], sequence-to-sequence models based on Recurrent Neural Networks were the state-of-the-art models to deal with context in a temporal sequence. However, they are computationally expensive as their architecture prevents parallelisation, and dealing with long-range dependencies is a challenge. [28] Transformers, on the other hand, tackle these problems. What makes Transformers so special is their concept of self-attention. A. Vaswani et al.'s explains self-attention as: "Self-attention, sometimes called intra-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence."[27] Although self-attention is a compelling concept, it was still limited to a fixed-length input. [28]

Transformers are built upon encoders and decoders. The architecture and their functionality are the same except that the decoder takes additional input from the encoder. Figure 2.1 displays a high-level view of a Transformer's encoders and decoders and how they are coupled.

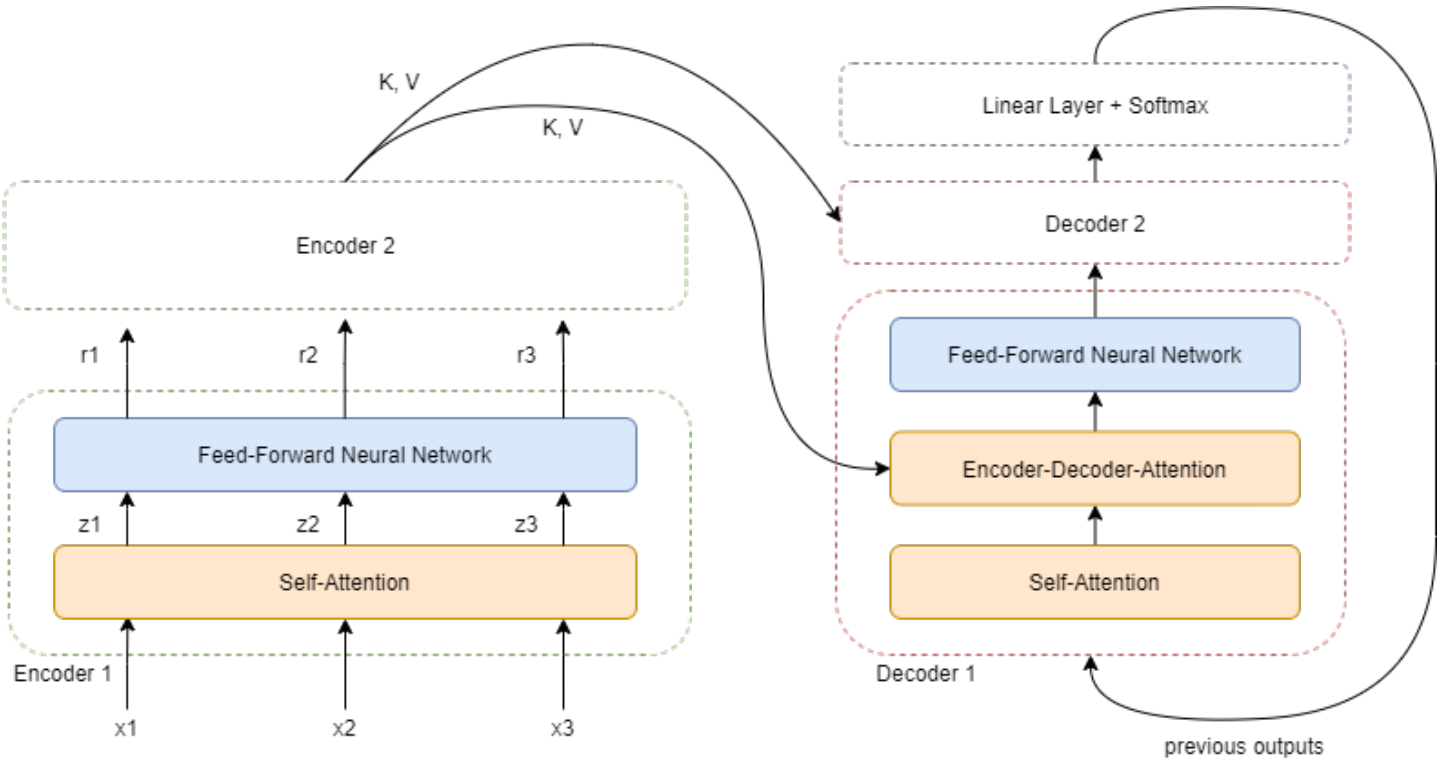


Figure 2.1.: High-level view of how input is processed trough a encoder-decoder Transformer.

In a Transformer-only network, the first decoder and encoder also need to map the input to an embedding representation. The number of encoder and decoder can vary depending on the chosen architecture, but the initial proposal worked with six of them. The first encoder takes the input representing embedding x_i and computes its corresponding internal values. It then runs the self-attention computation and passes the resulting z_i into a feed-forward neural network. The output r_i of each encoder is then passed to the next encoder as inputs. The matrices keys K and values V are then fed into every decoders' encoder-decoder-attention layer, enabling them to focus on essential parts of the input sequence while decoding. [28] The self-attention layer of each decoder is used to process the made outputs. The output of the last decoder is then passed into a linear layer that is a simple, fully connected neural network. After passing that output through a softmax layer, the final vector represents the probabilities for each word in the vocabulary. The corresponding word with the highest probability represents the first word of the output. Finally, the whole output is embedded and represents additional input for the decoders next iteration to predict the next word for the sequence in the process.

Self-attention

The self-attention computation takes a query Q, keys K, and values V as input. Those matrices are computed by multiplying the embedding matrices X with the corresponding internal weight matrices W_Q, W_K or W_V. Defined in Equation 2.1. Let I be $I \in \{Q, K, V\}$

$$I = X \times W_I \tag{2.1}$$

The self-attention computation is defined in Equation 2.2:

$$attention(Q, K, V) = softmax(Q \cdot K^T)V \tag{2.2}$$

Equation 2.2 is the classical dot-product (multiplicative) attention function, even though A. Vaswani et al. proposed a slightly different function that includes a scaling factor. The scaling factor is ignored for simplicity, as it is not important to understand the concept.

BERT

Bidirectional Encoder Representation from Transformers (BERT) [29] is a bidirectional implementation of Transformers XL [30]. Transformers XL are an advancement of the Transformer explained in section 2.2.1 to learn dependencies beyond the fixed-length limitations. BERT leverages the multi-layer bidirectionality to learn dependencies in both directions. It is pre-trained in an unsupervised - sometimes referred to as self-supervised - fashion. To pre-train, it uses Masked Language Modeling (MLM) and next sentence prediction. MLM enables BERT to learn to predict words within a sentence. This is achieved by masking 15 % of the words in pre-training according to the following rules:

- 80 % of the masked words are replaced with a mask-token.
- 10 % are replaced with a random word.
- 10 % are deleted and not replaced with anything.

This left BERT with the challenging task of predicting the masked word and checking if there is a word missing or an incorrect word in the sentence. The 20 % without mask-tokens are essential because, in the fine-tuning task, BERT will never see the mask-token.

In the sentence prediction task, pairs of sentences are fed to the network, and it has to predict if the second sentence is the correct next sentence. In 50 % of the pairs, the second sentence is replaced with a random sentence from the corpus for pre-training. [28] This training involves building an overall understanding of how sentences are dependent on each other.

2.2.2. Latent speech representation

Speech is a composition of phonemes, which describes a group of sounds that all have the same meaning-distinguishing function in a language. In theory, the number of phonemes per language is tiny. For example, in English, there are 44 phonemes, and in Spanish, 24. [31] In reality, speech waveforms, however, have a complex distribution with a high variance. This arises because of a wide variety of factors that influence the way people speak. They include dialects, accents, speaker identity, emotional state, surrounding sounds, etc. [32] The extraction of the smallest perceptible but distinct sound fragments from from these waveforms yields the so-called latent speech representations. Before 2017, attempts were made to model these by hand. However, by processing large amounts of unlabelled speech, these representations can be learned unsupervised. [32]

2.2.3. Quantization

As the latent speech representation is learned in an unsupervised way, one has to build a ground truth to calculate a loss to optimise the Transformer in training. This ground truth is created by quantization, which transforms the continuous latent speech representation into a discrete vector called quantized representation. A quantizer has one or multiple codebooks, which can be seen as dictionaries containing various discrete representations. The quantizer itself is nothing more than a mapping function that returns the nearest value from the codebooks for a specific continuous speech representation [33]. In wav2vec, this is done via product quantization. [1] This uses several distinct codebooks. Each speech representation is split into several subvectors that match the number of codebooks before quantizing them separately. In the end, the results are then concatenated into one single vector again. [33] To choose the representations from the codebook in a fully differentiable way, the Gumbel softmax is used. [1] [34]

2.2.4. wav2vec 2.0

The model of wav2vec illustrated in Figure 2.2 consists of a multi-layer convolutional feature encoder represented by the blue trapezoids. In the source code, this is also called a feature extractor. As input, it takes raw audio waves X and outputs latent speech representation Z. It does this for T time-steps using a sliding window of 25ms with a stride of 20ms. The outputs Z of the feature encoder are discretised to a finite set of speech representations using the quantization module described in subsection 2.2.3. On the other hand, Z is partially masked and fed to a Transformer, which is built similarly like BERT is. [29] The Transformer then builds contextualized representations C over the whole input sequence X. [1]

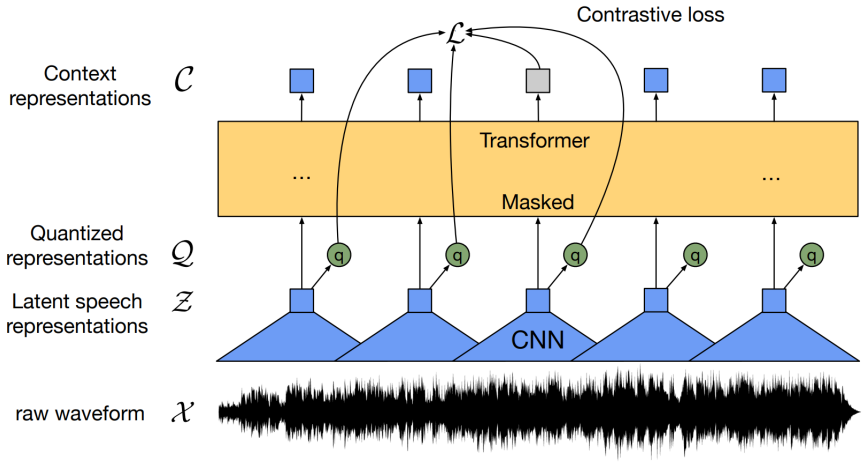


Figure 2.2.: High-level architecture of wav2vec 2.0 and how context representations are learnt from raw waveforms.

Wav2vec is pre-trained similarly to BERT. To pre-train the model, some randomly selected parts of the latent speech representations Z are masked and fed into the Transformer. Masking is done in three different ways simultaneously but not in equal distribution:

- Replacing a time-step with a mask-token.
- Replacing a time-step with a random different time-step.
- Cutting out a time-step with no defined mask.

These modified or deleted time-steps then have to be predicted by the Transformer. To verify the resulting context representations C, the model calculates a contrastive loss between them and the quantized representations Q, which is then used to optimize the Transformer. When calculating the contrastive loss, it tries to minimize the distance between C and Q while maximizing the distance to a set of distractors sampled from other masked time steps. For this task to work, it needs the codebook to represent positive and negative examples. For this purpose, a diversity loss is applied, which leads to the representations of the codebooks being used as equally often as possible.[1]

2.2.5. wav2vec 2.0 XLSR

Wav2Vec XLSR builds upon wav2vec 2.0; it follows the same architectural choices of their previous work in A. Baevski et al. [1]. The main goal of XLSR was to learn speech representations across multiple languages. The data to pre-train the model was collected from Mozilla's Common Voice (Common Voice) [35], BABEL that includes several African and Asian languages and the Multilingual LibriSpeech that includes audiobooks. Those datasets were integrated into one big dataset. In Figure 2.3 it can be seen how the latent speech representations are shared between languages. The most significant benefit of sharing these representations is the possibility to use language features from another language without pre-training the model again on a new language.

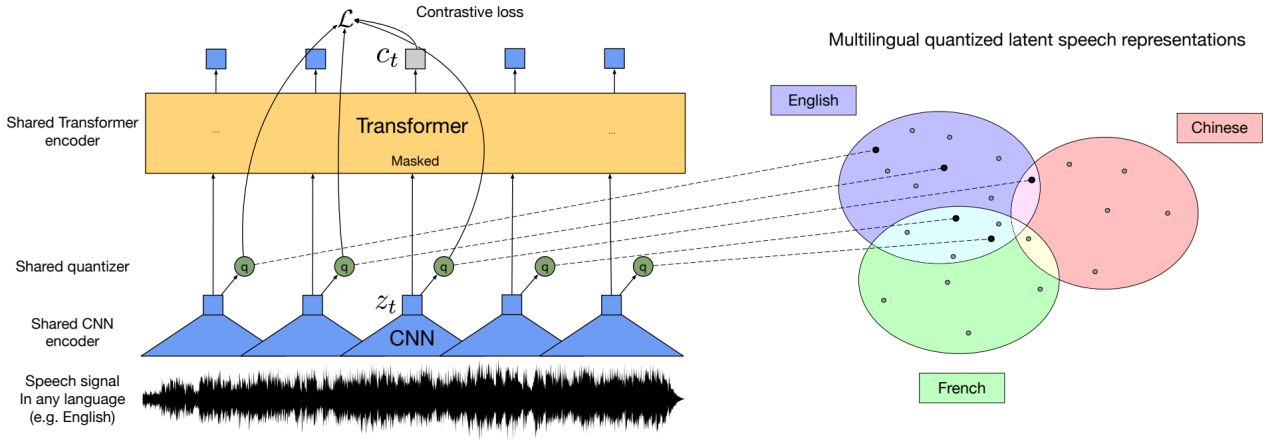


Figure 2.3.: Wav2vec XLSR and how speech representation are shared across languages.

The model used in this thesis and the biggest one released by Facebook AI is called wav2vec XLSR-53. It includes a pre-trained model on 53 languages covering 56'000 hours of audio. As there was not an equal distribution across the languages in pre-training, data-rich languages were penalized. [15]

2.3. Accent / Dialect Identification

DID refers to identifying the different dialects in a specific language spoken by natives. In addition, AID however can involve non-native-speakers who are greatly influenced by their maternal language. The task of identifying a dialect or accent is a subproblem in the category of Language Identification (LID). However, it is considerably more challenging as it detects differences in the phoneme space and how words are pronounced within the same language. [36] However, some differences exist when considering dialects: Dialects occur in small areas, and some spread over vast geographical regions or even continents. For example, the Arabic language can be distinguished into five different dialects: Egyptian (EGY), North African (NOR), Gulf or Arabian Peninsula (GLF), Levantine (LAV), and Modern Standard Arabic (MSA). However, as stated by A. A. Najim et al.: "An objective comparison of the varieties of Arabic dialects could lead to the conclusion that Arabic dialects are historically related, but not synchronically, and are mutually unintelligible languages like English and Dutch." [9] Therefore,

it must be considered that these dialects can be considered sufficiently distinct. Hence the distinction is more in line with LID than DID.

As described in Section 1.3, accents are more specific to how words are pronounced. AID is therefore even harder than DID as it is only possible to focus on the differences in pronunciation. Even for humans, it is challenging to distinguish between accents. In an experiment [37] participants had to distinguish 14 British dialects from telephone conversations they achieved an accuracy of 58 %. For humans, it is particularly difficult to distinguish between accents or dialects from regions where they never lived.

In practice, DID or AID can be used to improve ASR systems to transcribe text more accurately. The ultimate goal would be to detect the dialect or accent of an audio sequence and then distribute it to an ASR system optimised for that specific dialect, thereby achieving better results.

2.4. Metrics

To verify the quality of a classifier, it is crucial to choose the right metrics for the use case, as each metric makes a different statement about the classifier. The metrics used in this thesis are explained further in the following sections.

Accuracy

Accuracy is one of the most known metrics to classify if something is good. However, the problem with accuracy is that, in an unbalanced setting, it is easy to achieve high accuracy by always predicting the class with the higher distribution. What accuracy shows is all the correct predictions to the total predictions; more formally, accuracy is defined in Equation 2.3.

$$Accuracy = \frac{TruePositive + FalsePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative} \tag{2.3}$$

Precision And Recall

Precision, also called Positive-Predictive-Value, is a metric that summarises how many predicted outcomes are actually correct. As defined in Equation 2.4, it is calculated per class by setting all True-Positives in relation to the actual results. [38]

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2.4}$$

Recall, also called True-Positive-Rate or Sensitivity, on the other hand, makes a statement on how many of a specific class are recognised as such. As defined in Equation 2.5, it is calculated per class by setting all True-Positives in relation to the predicted results. [38]

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.5}$$

F1-Score

It is challenging to optimise Precision and Recall simultaneously as they tend to change in opposite directions. The F1-score resolves this problem. As defined in Equation 2.6, it is calculated by weighting Precision and Recall equally and building a harmonic mean, resulting in a metric that can be used to optimise both simultaneously. [38] Therefore, it is often used to evaluate the quality of classifiers and, in this thesis, for comparing the experiments.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.6}$$

Multi-class F1-Score

In a multi-class environment, other factors must be included to calculate the F1-score. The first and most basic way to calculate an F1-score, in this case, is macro-averaged, meaning that one calculates the F1-score per class individually, as explained above and takes the mean over all results. However, this also ignores a possible unbalance in the distribution of samples per class.

If the ignorance of this unbalance is not wanted, the F1-score can be calculated in a weighted-averaged manner. When averaging the individual F1-scores together, each score gets multiplied by the number of samples in that class first.

The last way of calculating an F1-score is micro-averaged. This is different from the other two. It works by first calculating the micro-averaged precision and the micro-averaged recall over all samples and then calculating the F1-score on that base. Thus simulating a kind of binary setup again. What is special for this case is that the following statement is always true for a micro-averaged F1-score: $precision_{micro} = recall_{micro} = F1_{micro} = accuracy$.[39].

Which type of calculation to choose depends on the standpoint one wants to support. This thesis aims to use the F1-score to make a statement about how good a model would generalise. Since in the corpora used, some individual classes, unfortunately, have very few samples, the macro-averaged F1-score is chosen. This means that the imbalance is not considered, which corresponds more to the actual general reality.

Macro Averaged Mean Absolute Error

Some of the experiments have classes that have a logical order in relation to each other. To assess how far the average error is from the correct classes, another metric is needed in addition to the F1-score. The Mean Absolute Error (MAE) measures the average error across all predictions in absolute terms. Let h be the classifier, Te the data samples and y the corresponding correct classes for each sample; the MAE is defined as follows.

$$MAE(h, Te) = \frac{1}{|Te|} \sum_{x_i \in Te} |h(x_i) - y_i| \tag{2.7}$$

However, this metric in Equation 2.7 is not really suitable for an unbalanced dataset because it treats all predictions equally regardless of which class they belong to. Therefore, classes that have more representations in the dataset receive a stronger weighting.

Thus, the MAE can be macro-averaged by splitting the predictions by its true class, calculating the MAE separately and taking the mean of the results.[40] Be C the distinct classes and Te_j the samples whose true class is C_j; the Macro Averaged Mean Absolute Error (MAEM) is defined in Equation 2.8.

$$MAEM(h, Te) = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{1}{|Te_j|} \sum_{x_i \in Te_j} |h(x_i) - y_i| \tag{2.8}$$

2.5. Plots

Confusion Matrix

The main plot for comparing experiments in this thesis will be the confusion matrix. It is helpful for unbalanced data and classifiers with more than two classes, as it can clearly show where the classifier is making mistakes. A confusion matrix summarises the number of correct and incorrect predictions for each class.[41] The scores in the confusion matrix can be normalised by plotting the scores relative to each classes size. This is particularly useful for unbalanced datasets and is, therefore, most commonly used in this thesis.

Precision-Recall Curve

The Precision-Recall Curve (PR-Curve) represents the qualities of the classifier graphically. While the Receiver Operating Characteristic Curve (ROC-Curve), which is also widely used, summarises the trade-off between the recall and the false-positive-rate. The PR-Curve, however, does so for the trade-off between the recall and the precision. This is particularly useful in unbalanced datasets. The y-axis shows the precision, and the x-axis the recall for different threshold values. For example, a No-Skill classifier would print a horizontal line with the value of the ratio of positive cases in the dataset. A perfect classifier would print a point in the top right corner(1, 1), and a skilful classifier is represented by a curve that bows towards that point.[42]

2.6. Transfer learning

In general, Transfer learning refers to the use of existing knowledge in a different domain. For example, in machine learning, the trained capabilities of one model are used on different related tasks. This enables models trained on large amounts of data to be reused and trained on a task where smaller datasets are more common. In terms of applying Transfer learning in this thesis, the capabilities from wav2vec in recognising features in natural speech are used to classify and distinguish accents, age and the sex of the speaker.

3. Experimental Setup

3.1. Objectives

The main objective of this thesis is to investigate the capabilities of wav2vec in the case of transfer learning, in particular, the applicability of wav2vec to distinguish accents within one language. Three different classification models based on wav2vec will be trained: AID, Age Identification and Sex Identification. It will be tested how the training volume affects the classifier's success in each domain. In addition, the performance of the trained AID models will be examined based on their performance at different lengths (e.g. one second) of audio samples. Furthermore, it is investigated whether this success can be further improved by taking a most voting over several short samples.

3.2. Corpora Selection

The corpora used had to meet several requirements, as well as general public availability. The following requirements had to be met by a corpus to be eligible for the experiments:

- speaker independence between train and test sets
- metadata for accents or dialect
- length of audio samples approximately five seconds or longer
- accents are all within the same language

Common Voice

Mozilla's Common Voice (Common Voice) is a crowdsourced corpus designed to help machines understand how people communicate. It currently includes 60 languages with about 7'000 validated hours. [43][35] People can record their voice, freely with or without creating an account, directly on the Common Voice homepage. The recording is reviewed by other users and eventually added to the corpus. If a speaker has created an account, additional demographic data can be added to that speaker's recording. Demographic data includes, for example, accent, age or gender. The recording itself is based on reading a sentence into a microphone. The corpus thus provides the required metadata, is speaker-independent, and most audio samples are about five to six seconds long. Depending on the language, there is a different amount of data available. Therefore not every language is suitable. For the experiments, only three of the 60 languages are used. The original corpus is divided into three parts: dev, train and test. Since there is limited labelled test-data for the English and Spanish corpus, these test-data were augmented with the dev-data. This was eligible because it was verified that there was no data leakage in speaker independence between the dev-data and the train-data.

English Common Voice

The extracted English dataset consists of six accents with sufficiently labelled data. The constellation shown in Table 3.1 refers to the subset of the entire English Common Voice corpus. In principle, there are nine accents in the English dataset, but only six accents had more than 5'000 labelled samples. If possible, double the number of the smallest accent was taken for all accents to get more variety in the dataset. The surplus was removed randomly for each class.

Accent	train samples	train duration	test samples	test duration	Avg. Duration
australia	12'360	19h 09m 37s	78	07m 54s	5.58s
canada	12'360	19h 24m 10s	141	15m 37s	5.66s
england	12'360	18h 23m 41s	431	43m 37s	6.07s
indian	12'360	19h 43m 08s	805	1h 20m 29s	5.23s
scotland	6'180	11h 01m 32s	24	02m 32s	6.42s
US	12'360	18h 50m 16s	1'575	2h 34m 47s	5.53s
total	67'980	106h 32m 24s	3'054	5h 04m 56s	5.75s

Table 3.1.: English dataset containing six accents with *autralia* and *scotland* having less than 80 test-samples.

Spanish Common Voice

The extracted Spanish dataset consisting of seven accents was extracted the same way as the English corpus. It had initially nine accents, but two had to be removed because they had less than 4'000 samples in the train-data. The remaining seven accents can be geographically grouped into Spain and Hispanic America. In Spain, there are: *nortepeninsular*, *centrosurpeninsular* and *surpeninsular*, whereas in Hispanic America are: *mexicano*, *caribe*, *andino* and *rioplatense*. The resulting corpus can be viewed in Table 3.2.

Accent	train samples	train duration	test samples	test duration	Avg. Duration
andino	7'306	11h 19m 24s	655	1h 07m 16s	5.63s
caribe	5'058	8h 19m 08s	466	48m 17s	5.95s
centrosurpeninsular	5'437	7h 59m 01s	300	28m 38s	5.31s
mexicano	10'116	15h 52m 14s	1'082	1h 47m 47s	5.68s
nortepeninsular	10'116	15h 00m 45s	360	34m 03s	5.35s
rioplatense	7'476	11h 37m 25s	438	44m 33s	5.63s
surpeninsularr	10'116	13h 47m 22s	176	16m 52s	4.92s
total	55'625	83h 55m 19s	3'477	5h 47m 26s	5.50s

Table 3.2.: Spanish dataset containing seven accents with *caribe* and *centrosurpeninsular* having less than 5'500 train-samples.

German Common Voice

The extracted German dataset did not contain enough data to use for AID, as it only contains three accents with an unbalanced distribution. Nevertheless, it was suitable for age and sex classification, with the advantage of making it easier to perform sanity checks after training. Common Voice provides metadata for age and gender.

For the age, metadata from *teens* to *nineties* is provided. Table 3.3 displays the constructed age corpus. The data from *sixties* to *nineties* had to be merged into one class to get a balanced corpus.

Age	train samples	train duration	test samples	test duration	Avg. Duration
10 - 19	8'636	12h 07m 08s	262	24m 01s	5.07s
20 - 29	17'272	25h 16m 53s	582	53m 37s	5.28s
30 - 39	17'272	26h 03m 06s	453	43m 21s	5.44s
40 - 49	17'272	31h 04m 07s	269	26m 19s	6.47s
50 - 59	17'272	27h 07m 53s	280	29m 34s	5.67s
>= 60	9'703	17h 23m 09s	170	18m 52s	6.45s
total	87'427	139h 02m 16s	2'016	3h 15m 44s	5.73s

Table 3.3.: German age dataset containing seven classes with a balanced distribution.

As Common Voice collects the metadata under the term gender, the classes were originally *female*, *male* and *other*. However, the goal is to identify the biological sex by voice only. Therefore, only the classes *male* and *female* were extracted, knowing that some people could identify as the other gender. The resulting sex corpus in Table 3.4 is also very unbalanced, which is particularly noticeable in the area of test-data. Apart from that, the corpus meets our specifications by far.

Sex	train samples	train duration	test samples	test duration	Avg. Duration
Female	22'114	35h 34m 58s	246	24m 59s	5.80s
Male	44'228	70h 58m 26s	1'745	2h 48m 34s	5.78s
total	66'342	106h 33m 24s	1'991	3h 13m 33s	5.79s

Table 3.4.: German sex dataset containing two classes, with *female* having only 246 test-samples.

ADI5

The five classes Arabic Dialect Identification (ADI5) corpus consists of the five dialects: EGY, GLF, LAV, MSA, and NOR, each providing more than ten hours of speech broken down in Table 3.5. The corpus is a collection of two corpora from the Multi-Genre Broadcast-2 and 3 challenge held at 2016[1] and 2017[2] IEEE Workshops. [8] The data is based on TV recordings from the international Arabic news channel Al Jazeera and different YouTube channels. As the Common Voice corpora, the ADI5 corpus is split into dev, train and test. Since the corpus is only used to recreate other results, the train-data was augmented with the dev-data as was done for the results to be reproduced. This was done without knowing if there were any data leakage between the dev-data and the test-data.

[1] 2016 IEEE Workshop on Spoken Language Technology.
[2] 2017 IEEE Automatic Speech Recognition and Understanding Workshop

Dialect	train samples	train duration	test samples	test duration	Avg. Duration
EGY	3'492	14h 23m 14s	302	1h 59m 27s	15.54s
GLF	3'138	12h 07m 34s	250	2h 04m 35s	15.09s
LAV	3'466	12h 16m 08s	334	2h 00m 02s	13.52s
MSA	2'502	12h 23m 48s	262	1h 56m 26s	18.67s
NOR	3'560	12h 28m 15s	344	2h 06m 38s	13.45s
total	16'158	63h 38m 59s	1'492	10h 7m 8s	15.25s

Table 3.5.: ADI5 dataset containing five classes with a balanced distribution.

3.3. System Selection

At the beginning of this thesis, it was tried to extend the wav2vec XLSR model with a fully connected layer. Since wav2vec's output depends on the length of the input data, the idea was to average the output features so that our model could work with flexible input sizes. However, during training with the ADI5 corpus, it became clear that there was a bug in the code as the loss was not decreasing. By then, a member of the Arabic Machine Learning (ARBML) community successfully implemented a classifier based on wav2vec. [44] He published his model, named Klaam, on Hugging Face with an accuracy of 83.78%. [45] His implementation differed slightly from ours in one important feature, which will be described in more detail shortly. Nevertheless, it showed that the idea of using wav2vec's XLSR model for speech classification has real potential.

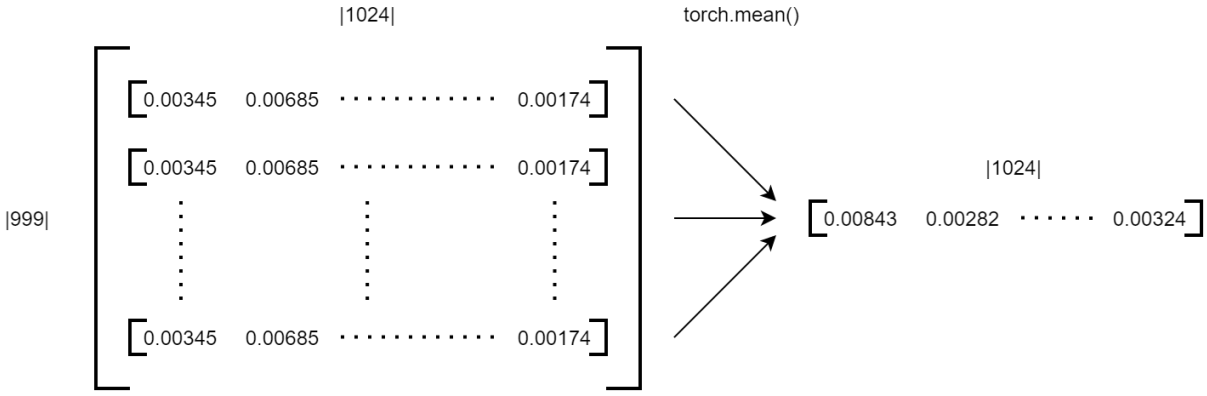


Figure 3.1.: Visualized dimension reduction of wav2vec's output using torch.mean().

In order to better understand the adaptations to the model, the outputs of the XLSR model must first be discussed. The output of wav2vec is a tensor with three dimensions: batch-size, number-of-output-features and number-of-layers. The batch-size can be ignored for better visualisation. What is interesting are the other two seen in Figure 3.1. The number of output features refers to the length of the input given to the XLSR model. For example: when the input has a length of twenty seconds, the number of output features is 999, and for ten seconds, it is 499. The number of layers is static and is 1'024. The handling of these variable sizes is done differently by the Klaam model and ours. As shown in Figure 3.2, the Klaam model reduces the output to a layer with the size 128, still carrying the dimension of output features, which is 999 as the model

was trained with twenty-second inputs. It then uses a Hyperbolic Tangent activation function and increases to a layer of the size 128*999 by concatenating the two dimensions into one large vector, leaving only the dimensions of the batch-size and the number-of-layers. With that, it reduces to a layer with the size of the number of classes to predict. This process makes the whole model dependent on the size of the model's inputs, which is undesirable. Therefore, our model eliminates the dimension of output features before it forwards the input to the classification layer using torch.mean(), which averages the output features into one vector. The resulting tensor is then put into a layer of the size 1'024, an activation function and another layer of the same size which gets reduced to the number of classes to predict.

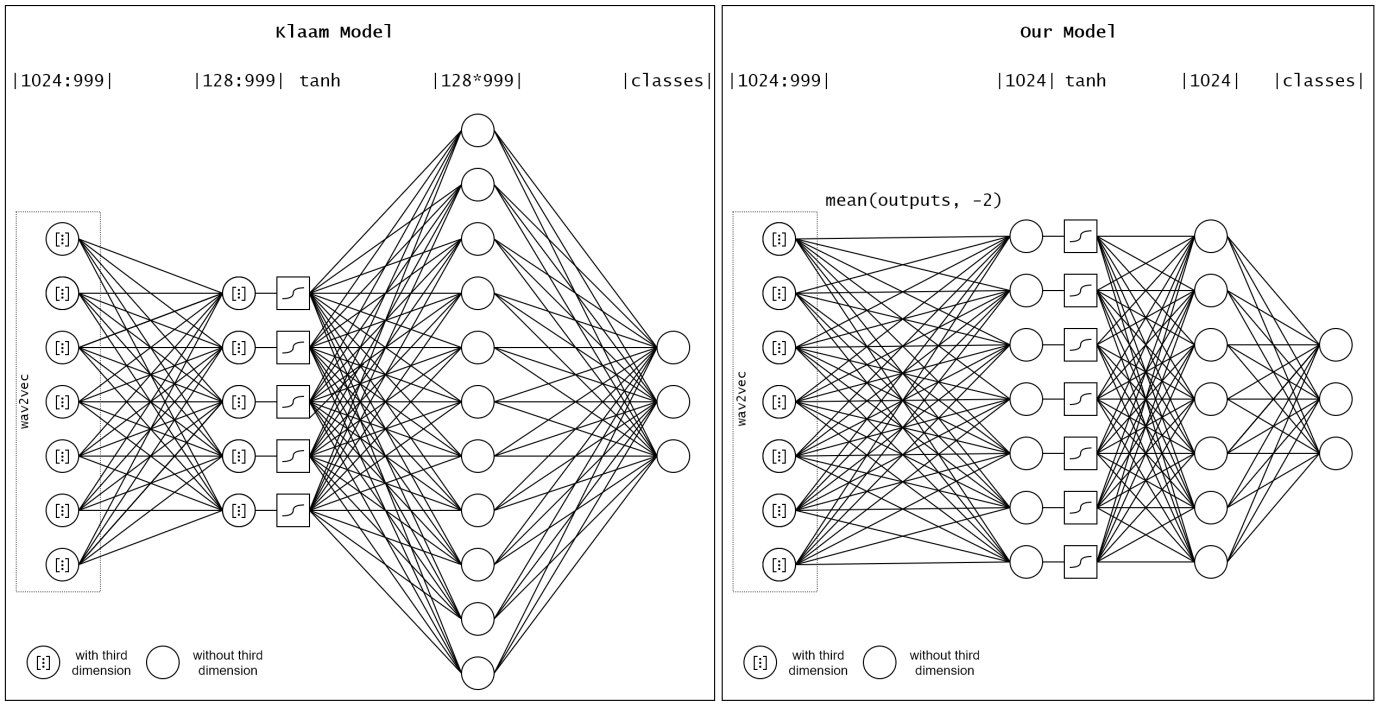


Figure 3.2.: Comparison between Klaam's classifier which concatenates the dimensions into one vector, and our classifier where the dimensions are reduced by averaging.

To verify if the changes were eligible. Our model was trained with the ADI5 corpus described in Table 3.5 to reproduce the accuracy of the deployed Klaam model. It was trained for ten epochs and matched the given figures with an accuracy of 84.25 % and an F1-score of 84.17 %. With a training time of 22 hours and 38 minutes, the total training time was reduced by 36 %. Therefore, our model was used for all experiments.

3.4. Metrics and Evaluation Tool

The primary metric for measuring the quality of the experiments used in this thesis will be the F1-score described in Section 2.4. The MAEM is also used but only for evaluating the experiments about age identification as it is the only topic where the order of classes is relevant. To calculate the metrics, the python library scikit-learn is used, except for the macro averaging in the MAEM, which had to be implemented by ourselves.

To log the metrics and diagrams, Weights & Biases (W&B) was used. It is a library that allows for tracking, comparing and visualising machine learning experiments and also provides quick and easy integration from various frameworks such as Hugging Face.[46]

3.5. Training

Training is done on the domains of accent, age and sex. The input data from the train-data and test-data is prepared by a pre-processor, which resamples the data to 16 kHz, cuts it to the desired length and pads it if necessary. The data is then mixed and divided into batches of size 16. The Hugging Face trainer is used as a training pipeline. This uses Adam as an optimiser that optimises a cross-entropy loss. Each model is trained for eight epochs on the train-data with an initial learning rate of 0.00003, which adjusts during training. The feature extractor of wav2vec is frozen so that the optimiser does not adjust its weights. After each epoch, the model is evaluated on the test-data, the metrics are calculated and transferred to W&B, where they are monitored. In the end, the final model is saved to a local directory.

3.6. Experiments

Four different main experiments are conducted. The first experiment is AID; it focuses on the different amounts of training data and the capabilities of wav2vec to learn on small datasets. The Sample Length Evaluation (SLE) experiment investigates what length an audio sample needs to be successfully classified. In addition to the SLE experiment, a most voted prediction is tested. It explores if the results of SLE can be improved if multiple parts of one file are classified on their own and pulled together for one classification. To further explore the capabilities of wav2vec, the last two experiments go in a different direction and try to identify the age or sex of the speaker. Each experiment is repeated three times.

3.6.1. Accent Identification

The first experiments revolve around AID. Since the ultimate goal of wav2vec is to enable speech recognition with limited data. [1] These experiments focus on how much training data is needed to create an efficient classifier.

Unfortunately, the experiments are limited in training data volume per class, as the aim was to train on balanced classes. The experiments conducted with the English and Spanish Common Voice corpus are listed in Table 3.6. In order to measure a correlation

between the amount of training data and success, the amount per class is increased by 1'000 per experiment up to the limits we have defined. Training is done with ten-second samples, as there are no samples longer than that. Shorter samples are padded to ten seconds.

ID	max length of input	volume per class	number of classes	language
AID-EN-1	10s	1'000	6	English
AID-EN-2	10s	2'000	6	English
AID-EN-3	10s	3'000	6	English
AID-EN-4	10s	4'000	6	English
AID-EN-5	10s	5'000	6	English
AID-ES-1	10s	1'000	7	Spanish
AID-ES-2	10s	2'000	7	Spanish
AID-ES-3	10s	3'000	7	Spanish
AID-ES-4	10s	4'000	7	Spanish

Table 3.6.: Accent identification experiments with increasing training-volume per class.

3.6.2. Sample Length Evaluation

In a conversation between people, the speaker changes very often, and the individuals only say short, coherent sentences. Therefore, experiments are conducted to find out how accuracy[3] behaves with short segments of different lengths. For this purpose, the models from the accent identification experiment with the largest training volume are used. Each of the three models per language is evaluated with different sample lengths. Care is taken to use the same volume of test data as in the original Accent Identifications experiment. The experiments to be conducted can be seen in Table 3.7. Since the average sample length for both languages is about five to six seconds, the maximum sample length tested is five seconds.

ID	max length of input	number of classes	language
SL-EN-1	1s	6	English
SL-EN-2	2s	6	English
SL-EN-3	3s	6	English
SL-EN-4	4s	6	English
SL-EN-5	5s	6	English
SL-ES-1	1s	7	Spanish
SL-ES-2	2s	7	Spanish
SL-ES-3	3s	7	Spanish
SL-ES-4	4s	7	Spanish
SL-ES-5	5s	7	Spanish

Table 3.7.: Sample length evaluation experiments on models of AID-EN-5 and AID-ES-4.

[3]represented by F1-score

Most-voted prediction

Inspired by the sample length experiment, the classification could be improved by applying a most-voted prediction. As shown in Figure 3.3, an audio file is split into several parts of equal length. Each section is then classified by itself. After summing up all votes by class, the class with the most votes over all slices is selected as the final classification. If more than one class have reached the same number of votes, the decision between the classes with the same number is made randomly.

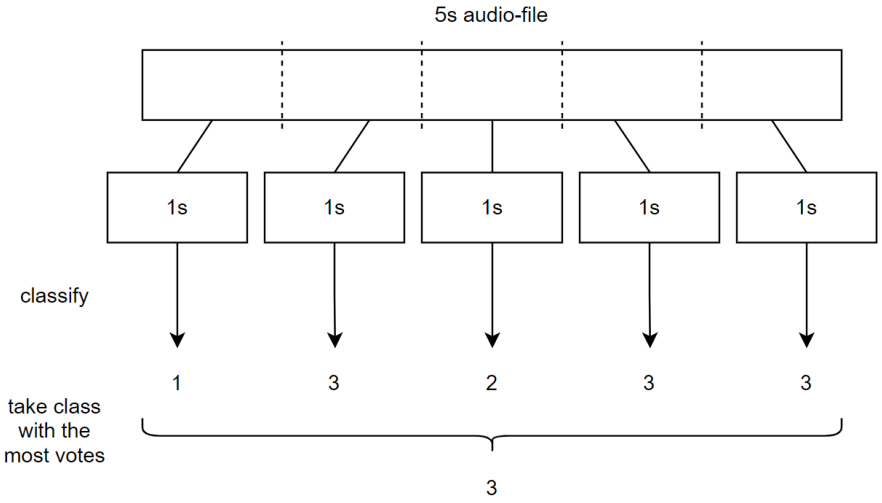


Figure 3.3.: Example of most-voted prediction

Experimentation is done with three different lengths of slices as listed in Table 3.8. The number of votes corresponds to the maximum number of votes cast for one audio file. There are few files longer than eight to nine seconds, so this is an acceptable loss. Each audio file is sliced as many times as possible until the maximum is reached. If the desired number of slices cannot be reached, the last slice is padded, and all other empty slices are ignored. Experimentation is conducted using the models trained in the AID experiments with the largest amount of training data.

ID	length of input	number of votes	number of classes	language
MV-EN-1	1s	8	6	English
MV-EN-2	2s	4	6	English
MV-EN-3	3s	3	6	English
MV-ES-1	1s	8	7	Spanish
MV-ES-2	2s	4	7	Spanish
MV-ES-3	3s	3	7	Spanish

Table 3.8.: Most voting evaluation experiments on models of AID-EN-5 and AID-ES-4.

3.6.3. Age Identification

To explore the limits of wav2vec in speech classification, experiments are also conducted that are more unconventional than classical AID. In these experiments, the suitability of wav2vec as a classifier for age identification is examined. It is attempted to identify the age of a person based on their voice. The goal is to recognise the age in ten-year groups except for people who are over 60, which should be classified as such.

Apart from the corpus, the experiments are similar to those described in Section 3.6.1. However, one significant difference is that the severity of a misclassified class can be taken into account. Since age is a continuous sequence, the severity of the error can vary as it is much better to estimate a person in his thirties as being in his twenties than to estimate him in his sixties. Therefore, in addition to assessing the quality of the experiment with the F1-score, the MAEM is also used to assess the severity of the error. The four corresponding experiments are documented in Table 3.9.

ID	max length of input	volume per class	number of classes
AGE-1	10s	1'000	6
AGE-2	10s	2'000	6
AGE-3	10s	3'000	6
AGE-4	10s	4'000	6

Table 3.9.: Age identification experiments with increasing training-volume per class.

3.6.4. Sex Identification

Another experiment that addresses the possible limitations of wav2vec as a classifier revolves around sex identification. In these experiments, the possibility of identifying the biological sex of a person by their voice is tested. From a human perspective, this should be easier to accomplish compared to age identification.

The classification is a so-called binary classification, as there are only two classes to choose from. As with the accent and age identification experiments, the aim is to evaluate the relationship between the amount of training data per class and the quality of the resulting classifier. The four experiments that will be conducted can be seen in Table 3.10.

ID	max length of input	volume per class	number of classes
SEX-1	10s	1'000	2
SEX-2	10s	2'000	2
SEX-3	10s	3'000	2
SEX-4	10s	4'000	2

Table 3.10.: Sex identification experiments with increasing training-volume per class.

4. Results

In this chapter, the results of all experiments are presented. Based on the initial results, further analyses are carried out to draw better conclusions. Where possible, the results are checked for plausibility by taking a closer look at the misclassified samples and conducting experiments with self-generated data.

Each experiment was repeated three times to assure certain stability of the results. Some experiments build on previous ones by evaluating trained models under additional conditions, such as shorter samples. To compare the performance of the different experiments in a topic, the metric of the macro-averaged F1-score is used, as the test-data sets are not well balanced, and therefore accuracy is not suitable. The listed F1-Scores represent the average of all three repetitions of an experiment.

4.1. Statistical Significance

To assess the significance of the different results obtained, the student's t-test provided by scipy was applied. The corresponding results can be found in the Appendix A.3. They show that the differences measured from the most successful experiment to all other experiments are mostly not significant. One reason for this is that with only three repetitions of an experiment, the standard deviation can be relatively large. Therefore, the results of this thesis should be treated with caution, as more repetitions would have to be carried out to draw more reliable conclusions[1].

4.2. Accent Identification

The goal of this experiment was to investigate what performance is possible with different amounts of training data. The experiments were conducted for English and Spanish, with the English experiments having one more experiment because the corpus was larger. The statistics of all runs can be viewed in Appendix A.1.1.

[1]This would go beyond the time frame that was available for this thesis

4.2.1. English

Table 4.1 shows the comparison of the different trained models and their averaging F1-score described in Section 2.4.

ID	volume per class	average F1	standard deviation
AID-EN-1	1'000	0.346	0.026
AID-EN-2	2'000	0.363	0.014
AID-EN-3	3'000	0.37	0.003
AID-EN-4	4'000	0.367	0.013
AID-EN-5	5'000	0.396	0.021

Table 4.1.: Results of English Accent Identification experiments and their averaged F1 over all runs.

It can be observed that the more data is available, the better the models are performing. Even though the rise in performance is not that significant as the rise from wav2vec when fine-tuned with more data. Experiments from Facebook AI's researcher showed a clear improvement if more data for fine-tuning was used.[1] However, it is astonishing that the model can reach an F1-score of 0.346 with only 95 minutes of training data per class.

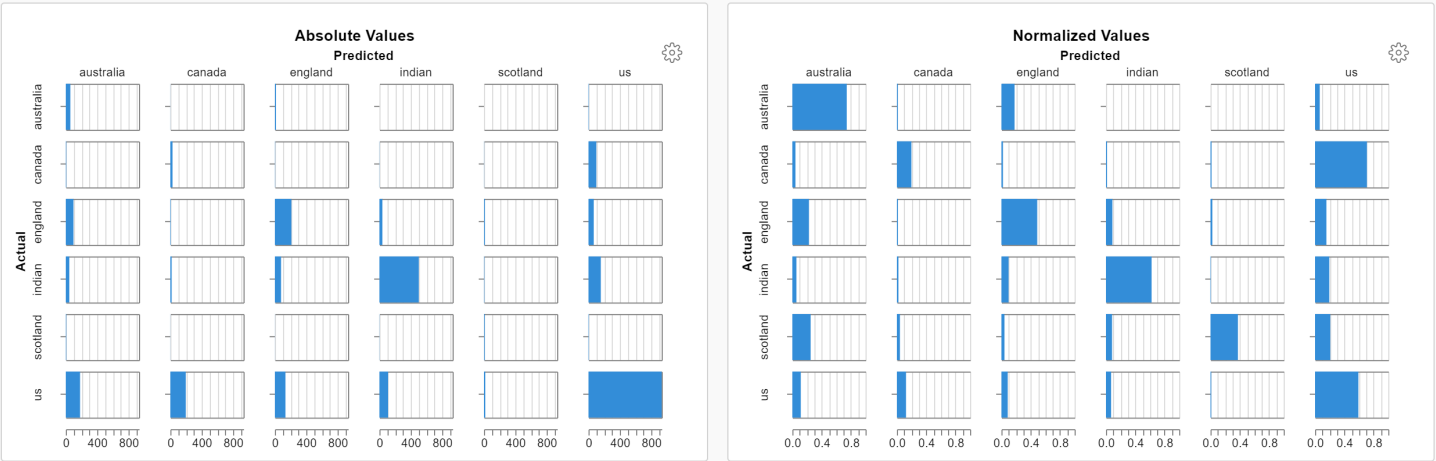


Figure 4.1.: Confusion Matrices of the best performing model of AID-EN-5 in absolute and normalized values.

Investigating further into the best performing model with an F1-score of 0.413, Figure 4.1 shows the confusion matrix in absolute numbers on the left side. On the right side, the normalized view presents that the model is confusing similar accents as a human would, such as confusing the *canadian* accent with the *us* accent or the *australian* accent with the *english* accent. Nevertheless, at the same time, being consistent in classifying the *indian* accent. The *indian* accent is more distinctive than the other accents as it is the only foreign accent in the dataset. Even for humans, it can be said that accents from non-native speakers are easier identifiable.

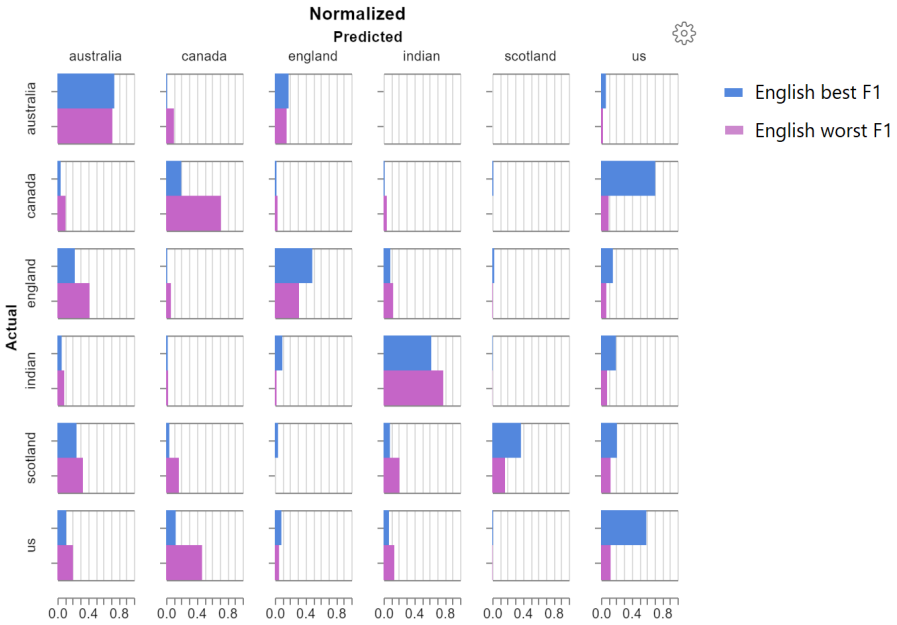


Figure 4.2.: Confusion Matrix of best AID-EN-5 and worst AID-EN-1 performing English model.

The fact that these confusions exist is preferably good, especially compared to the worst performing model with an F1-score of 0.313 in Figure 4.2 where the classification is more randomised, especially for the *us* accent. Nevertheless, it is also observable that the worst model focused on the *canadian* accent, whereas the best model focused on *us*. This is another indicator that *canadian* and *us* are hard to distinguish. In addition, the *english* accent is more often classified wrong as *australian* accent. However, the *indian* accent is classified correctly more often.

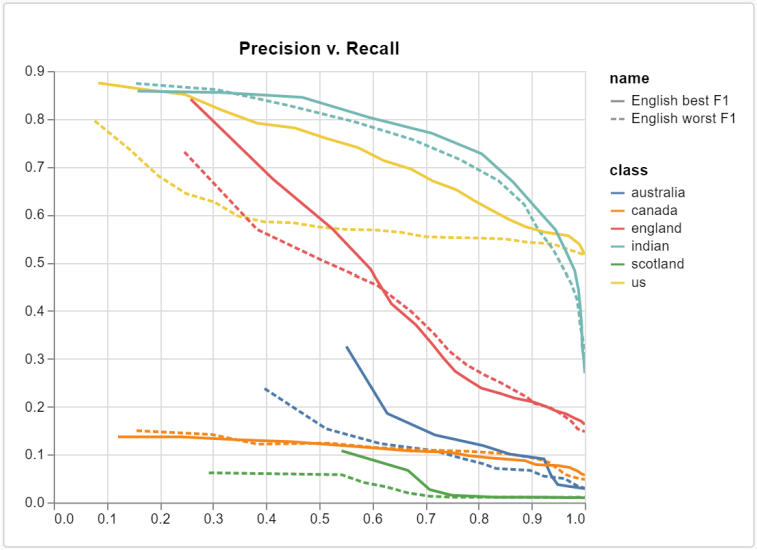


Figure 4.3.: Precision vs. Recall Curve of best AID-EN-5 and worst AID-EN-1 performing English model.

When comparing the models with the worst and the best F1-score in Figure 4.3, it becomes more apparent that the best model performs better on the *us* accent than the worst model. The *indian* accent seemed to be better classified by the worst model in Figure 4.2. In contrast, Figure 4.3 shows that the precision of the best model was still better. Training on more data, therefore, is still beneficial.

Since we are not native English speakers, it is difficult for us to distinguish between the accents of native English speakers. However, we can determine whether someone has a foreign accent. We are confident enough to determine whether someone speaks with an *indian* accent or not. Therefore, it was possible to check the miss-classifications in this class.

model	errors in total	correlating files	correlation
AID-EN-5-RUN-1	1'313	986	75 %
AID-EN-5-RUN-2	1'384	1098	79 %
AID-EN-5-RUN-3	1'697	1326	78 %

Table 4.2.: Absolute number of miss-classified files per run and how many of them are miss-classified in other runs (correlation).

Table 4.2 lists how many errors were made in the evaluation per model. The correlation tells how many incorrectly classified files can also be found incorrectly in one of the other models. On average, about 77 % of all files were miss-classified in more than one model. This could be an indicator that the quality of this data is not good.

indian predicted/actual	positive	negative
positive	502	153
negative	303	2'096

Table 4.3.: *Indian* accent: binary confusion matrix of all 3'054 files in test-data (F1:0.69, Precision: 0.77, Recall: 0.62)

For the *indian* accent, there were 153 false-positive classifications and 303 false-negatives, as seen in Table 4.3. 105 of the 153 files classified as *indian* were listened to more closely. These then were classified by hand, resulting in 55 with a clear *indian* accent. Table 4.4 lists the first 85 files with the highest confidence classified by the AID-EN-5-RUN-1 model. Surprisingly, when the model classifies a file as *indian* with a confidence greater than or equal to 90 %, the probability of the file having a wrong label is still 81 %. The overall F1-score for the prediction of the *indian* accent resulted in 0.69. Therefore, if an F1-score of around 0.7 is reached, the model's capability to detect miss-labelled data is good enough. Most of the miss-labelled files were intended to have an *us* speaker in them. The problem is that these labels come from user input, and anyone can enter what they want. The 303 false-negatives, on the other hand, of which 30 were listened to, almost all seem to have an *indian* accent. Therefore, these errors can be counted as real miss-classifications.

model's confidence	correctly classified as *indian*	total classified as *indian*	accuracy
$\geq$ 99 %	18	21	86 %
$\geq$ 95 %	35	43	81 %
$\geq$ 90 %	42	52	81 %
$\geq$ 80 %	52	71	73 %
$\geq$ 70 %	54	85	64 %

Table 4.4.: Accumulated miss-classifications of model AID-EN-5-RUN-1 errors in relation to its confidence. Accuracy represents the correctly classified files when errors are relabelled by hand.

Conclusion

More than 1'000 samples with an average length of five to six seconds are needed to train a more successful classifier. Especially when learning to distinguish similar accents, more data is needed. The F1-score increases as more data is available, but 5'000 samples per class is still not enough to achieve acceptable performance. As described in sanity testing, the quality of the test-data needs to be improved as the model correctly classified the files when its confidence was high. If the detection of incorrectly labelled test-data is taken as a benchmark, it could be suggested that from an F1-score of 0.7 per class, the model can detect miss-labelled data if the confidence is above 90 %.

4.2.2. Spanish

Apart from identifying accents in another language, the Spanish experiments have one more class to predict compared to the English accents. Therefore, the results shown in Table 4.5 also show that the F1-scores are much lower overall. However, they do not seem to improve much when the volume per class is increased in training. What stands out are the experiments AID-ES-2 with a volume per class of 2'000, which have a lower F1-score than the experiments with less training data and have a particularly high standard deviation. Even though the high standard deviation is mainly due to one particularly bad run, the other runs were not much better, as seen in Appendix A.1.2.

ID	volume per class	average F1	standard deviation
AID-ES-1	1'000	0.258	0.011
AID-ES-2	2'000	0.226	0.034
AID-ES-3	3'000	0.258	0.017
AID-ES-4	4'000	0.266	0.020

Table 4.5.: Results of Spanish Accent Identification experiments and their averaged F1 over all runs.

By investigating the corresponding confusion matrices from the worst run with 1'000 samples and the best run with 4'000 samples per class in Figure 4.4, it does not look very promising at first glance. However, when looking more closely at the miss-classifications of the best performing model represented in purple, it can be seen that even if *andino* is rarely classified correctly, it gets confused with the accents *caribe*, *mexicano* and *rioplatense* rather than *nortepeninsular* and *surpeninsular*. On the other hand, *nortepeninsular* gets confused with *centrosurpeninsular* and of all Hispanic American accents, only *mexicano*.

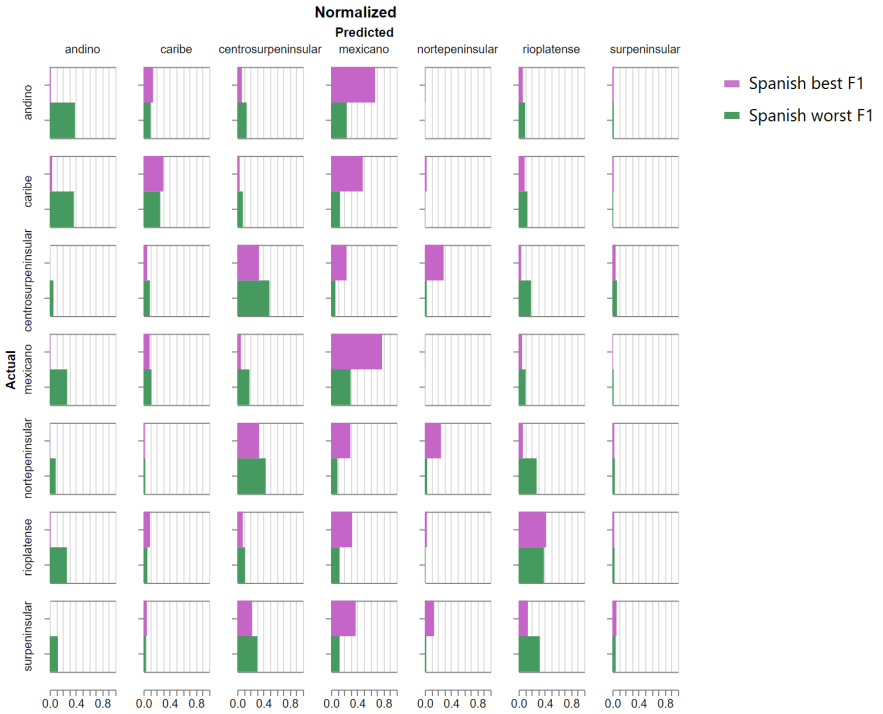
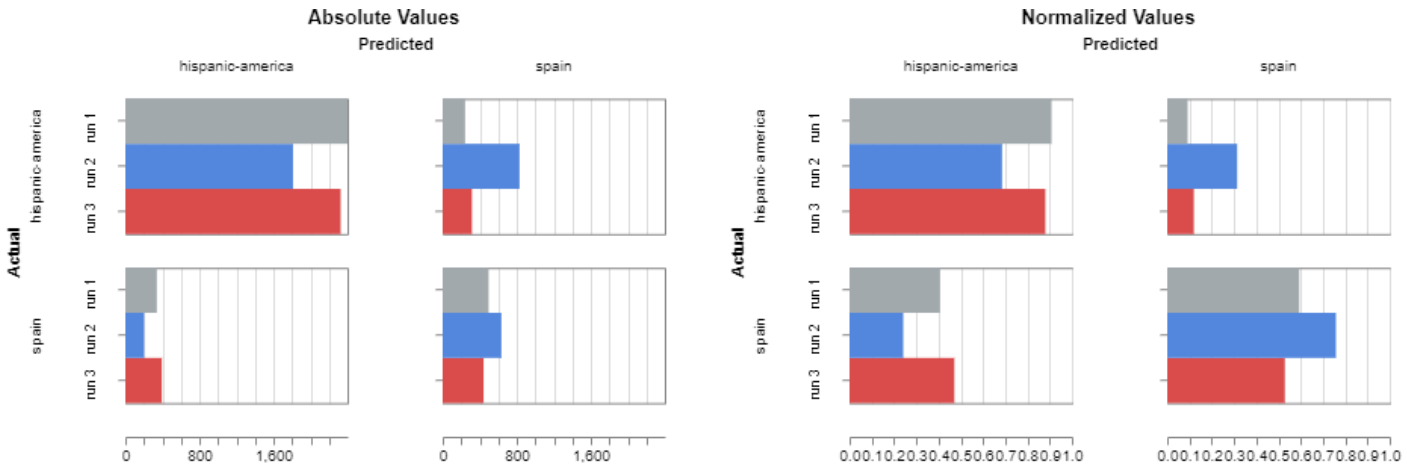


Figure 4.4.: Confusion Matrix of best AID-ES-4 and worst AID-ES-1 performing Spanish model.

This tendency led to the assumption that even though the classifier has difficulties identifying all seven accents, it can assign them to the two groups mentioned in Section 3.2: Hispanic-America and Spain. To check this, the evaluation of all three runs of the AID-ES-4 experiment were repeated and the resulting predictions grouped according to their geographical affiliation. The result was a classifier with an average F1-score of 0.712 and a standard deviation of 3.93 %. This is great, considering that the classifier was not trained on this particular task. It could be assumed that it would have been even better if trained on this binary task. However, this was not pursued. The detailed confusion matrices of the three runs can be seen in Figure 4.5.



(a) Absolute values of binary evaluation

(b) Normalized values of binary evaluation

Figure 4.5.: Confusion Matrices of binary evaluation of AID-ES-4.

Figure 4.5 shows that as in the original experiment AID-ES-4, the first run was again the best run with the fewest miss-classifications. However, the other two are directly inverted. It is also noticeable that the second run has a strange bias for Spain that is not noticeable in the others. This makes it difficult to say whether there is a correlation between success in training in the original setting and the evaluation in the binary setting.

Conclusion

The results of the Spanish experiments are a little worse than the English ones. The difference is that in the case of Spain, it was tried to classify three accents within one country. It also seems that the Hispanic American accents are more similar to each other. As the binary evaluation showed, the distinction between these two groups works much better. However, since we do not comprehend Spanish, it is difficult to analyse these results more deeply.

4.3. Sample Length Evaluation

In this experiment, it was evaluated how well a trained model can classify shorter sample lengths. All evaluations were done for each of the models trained in the AID experiments AID-EN-5 and AID-ES-4. A detailed listing of these results can be found in Appendix A.2.2 and A.2.3.

ID	length	average F1	standard deviation
SL-EN-1	1s	0.165	0.012
SL-EN-2	2s	0.241	0.014
SL-EN-3	3s	0.279	0.016
SL-EN-4	4s	0.301	0.020
SL-EN-5	5s	0.315	0.015
SL-ES-1	1s	0.144	0.014
SL-ES-2	2s	0.183	0.020
SL-ES-3	3s	0.207	0.014
SL-ES-4	4s	0.222	0.012
SL-ES-5	5s	0.225	0.015

Table 4.6.: Results of English and Spanish sample length evaluation experiments and their averaged F1 over all runs.

Table 4.6 shows the reached average F1-scores for a certain length of input data, classified by the different models and their corresponding standard deviation. It mainly shows that for both languages, Spanish and English, the classification was random for the 1-second samples. English had six different accents to classify and Spanish seven, resulting in 1/6 and 1/7 of hitting the correct class. The longer the sample, the better is the achieved F1-score. However, it is crucial to consider that the evaluation was made on models trained with a maximum sample length of ten seconds. Moreover, it could be possible that models trained on shorter lengths would have performed better, as wav2vec is specialised in building contextualised representations over longer sequences. In addition, our classifier was trained on the output from wav2vec of longer sequences, which makes it harder for our model to classify shorter samples.

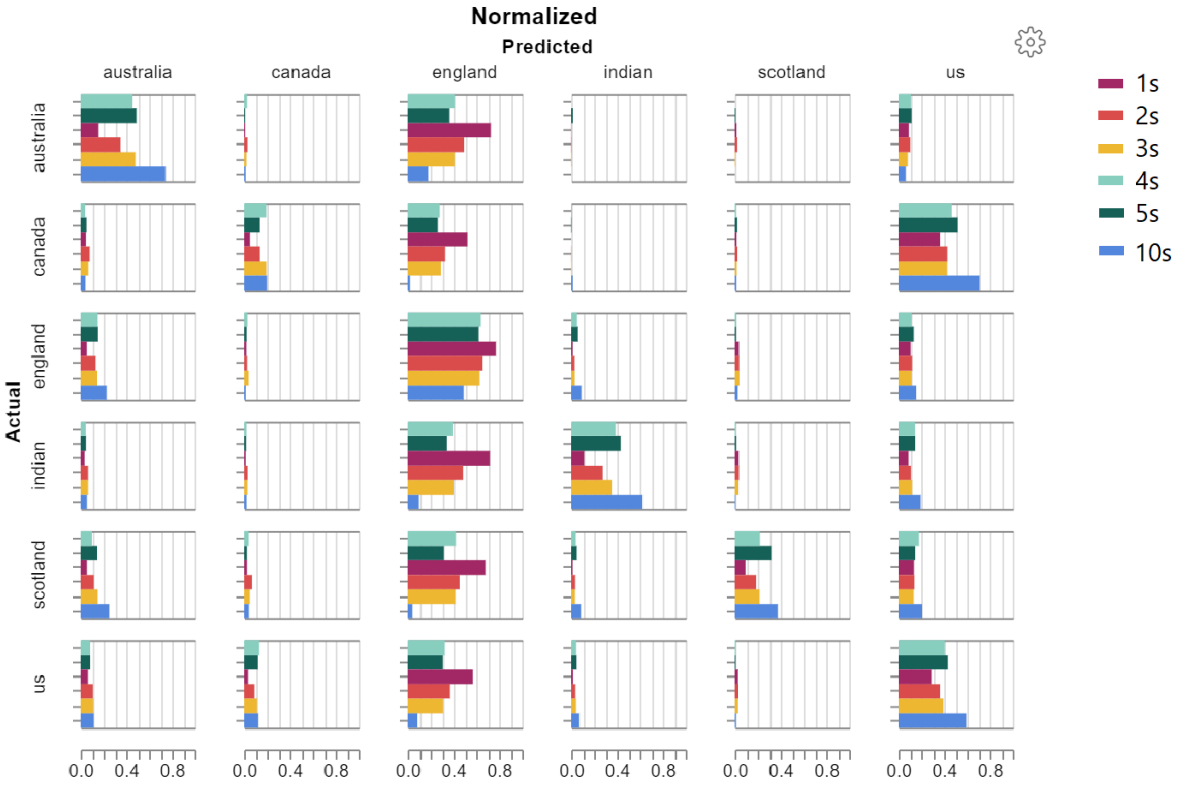


Figure 4.6.: Confusion Matrix English with best run for each sample length.

When taking a deeper look into the English classification results with the best performing models for each sample length. The confusion matrix in Figure 4.6 shows that the models are more biased to classify towards the *english* accent the shorter the samples are. If the 5s and the 10s evaluation are compared. For the 5s model, the *indian* accent is classified mostly wrong as *english*, whereas in the 10s model, the favourable choice was *us*. The favourable choice of *us* in the 10s reflects the dataset, as there are false labelled *indian* accented files in the *us* dataset. In terms of classifying almost everything as *english*, a possible explanation could be that *english* acts like a neutral accent, and it generalises best for all different accents. However, this theory has not been investigated further.

Similar behaviour can be observed in the Spanish Experiment. Although the previously proposed generalisation happens between *mexicano* and *nortepeninsular* instead of only one accent, as shown in Figure 4.7. However, the overall performance of the Spanish models are worse than those of the English ones and lay nearer the random threshold.

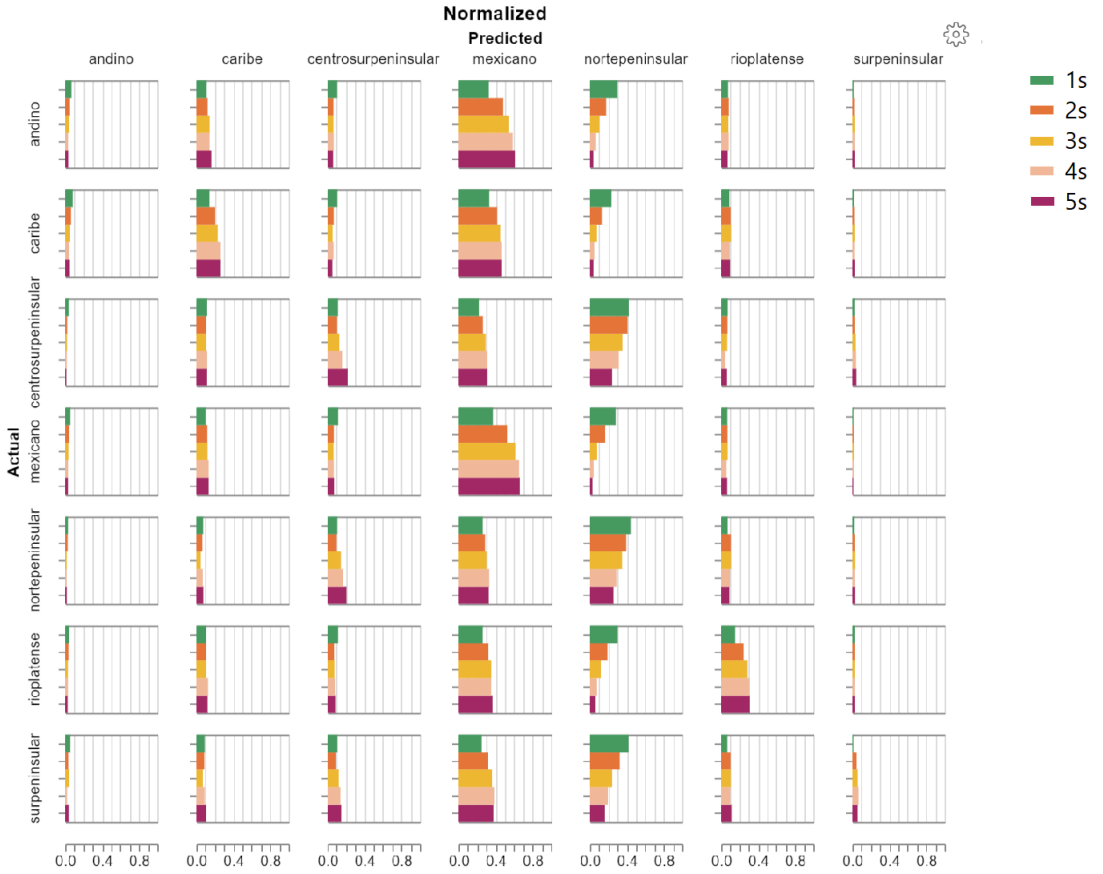


Figure 4.7.: Confusion Matrix Spanish with best run for each sample length.

Conclusion
To identify an accent with a classifier trained on a maximum of ten seconds, it is preferable to have input data as long as possible. On the other hand, we also see a trend of all trained models that are biased towards one accent if the length of the sample gets shorter.

4.3.1. Most voted

In the most voted experiment, the idea was to increase the accuracy by slicing one file into multiple samples, evaluating each of them and counting the votes. The final classification is therefore done on one file and not just on one slice. Finally, the class with the most votes per file is selected.

In Table 3.8, it can be observed that the evaluation is not getting better for the English models, especially for the 1-second slicing; it performs 40 % worse than the evaluation did on each sample on its own. On the contrary, the Spanish models performed better with the most votes evaluation. However, these numbers have to be considered as not too accurate as each time multiple classes have the same amount of votes; a random class is chosen. This especially gets problematic when comparing the results of the 3s-runs as the average length is under six seconds, resulting in only two possible votes.

ID	length	average F1	standard deviation	average F1 only 1 sample
MV-EN-1	1s	0.099	0.03	0.165
MV-EN-2	2s	0.221	0.023	0.241
MV-EN-3	3s	0.268	0.017	0.279
MV-ES-1	1s	0.157	0.016	0.144
MV-ES-2	2s	0.202	0.019	0.183
MV-ES-3	3s	0.214	0.015	0.207

Table 4.7.: Results of English and Spanish most voting evaluation experiments and their averaged F1 over all runs compared to the corresponding average score from Table 3.7.

Conclusion

The most-voted prediction does not benefit from better accuracy. Especially when files get longer, it is hard to classify a file when only two votes are possible. Furthermore, the performance is still coupled to the overall performance in classifying one short sample.

4.4. Age Identification

Looking at the age identification results summarised in Table 4.8, the documented F1-score ranges from 0.30 to 0.36. Even though the score seems to rise with increasing training data, the differences cannot be considered significant because the standard deviation is relatively high.

ID	volume per class	average F1	standard deviation
AGE-1	1'000	0.303	0.018
AGE-2	2'000	0.340	0.016
AGE-3	3'000	0.351	0.004
AGE-4	4'000	0.360	0.024

Table 4.8.: Results of Age Identification experiments and their averaged F1 over all runs.

However, when looking more closely at the confusion matrix of the best run in experiment AGE-4, shown in Figure 4.8, some interesting constellations in the normalised values stand out. The difference to all other experiments is that the age is continuous, which means that the order of the classes has a meaning.

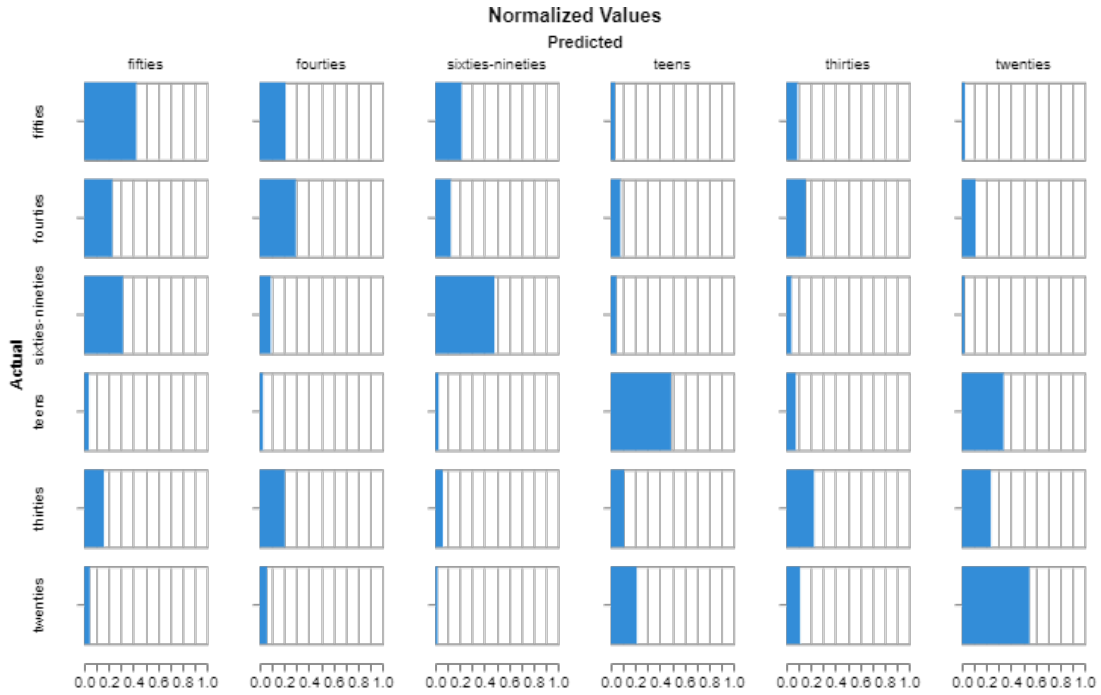


Figure 4.8.: Confusion Matrix of best AGE-4 performing age identification model.

The class *fifties* in the first row of the matrix in Figure 4.8 is mainly confused with the classes *sixties-nineties* and *fourties* and less with the classes *teens*, *twenties* and *thirties*. This observation can be made for most classes, leading to the Macro Averaged Mean Absolute Error (MAEM) defined as a metric for this experiment.

ID	volume per class	average F1	standard deviation
AGE-1	1'000	1.077	0.022
AGE-2	2'000	1.005	0.024
AGE-3	3'000	1.006	0.005
AGE-4	4'000	0.982	0.039

Table 4.9.: Results of Age Identification experiments and their averaged MAEM over all runs.

The measurements on the MAEM to the correct classifications, documented in Table 4.9, show that the actual average error is about one class apart. This puts the errors of the classifier into perspective and shows that it is not necessarily as bad as the F1-score alone makes it look. It is also important to point out that identifying a person's age just by their voice is not very easy, even for a human.

In the sanity testing, it was considered whether the sex of the speaker could influence the result of the age identification. For this purpose, the misclassified samples were examined more closely. It was found that over all three runs, a total of 3'632 samples were incorrectly classified. These were produced by a total of 508 different speakers. The sex, however, seemed not to influence the classifications, as can be seen in Appendix A.2.6. The distributions were almost similar.

For further investigations, all speakers that produced errors in all three repetitions were attributed by their absolute error. The ones with an error greater than or equal to four, were listened to more closely. The results did not yield much insight. However, it was suspected that speakers who had a bad microphone, spoke very loudly or very softly, or generally spoken more unclearly often fell into the *sixties-nineties* category even when they were very young. To check this, an attempt was made to classify own recordings of the same speaker saying the same sentence under different conditions. The tests were carried out with six different people, three of whom were *female* and three *male*. The age distribution for both sexes was two *twenties* and one *fifties*. The tested conditions were: normal, with background noise, far from the microphone and close to the microphone. The results in Table 4.10 do not confirm the assumption. However, they show that the setting influences the quality of the classification, whereby the settings of the different distances to the microphone had the greatest influence with the greatest MAEM.

condition	twenties	fifties	female	male	MAEM
normal	4	2	3	3	1.125
background noise	4	2	3	3	1.375
far from microphone	4	2	3	3	1.542
close to microphone	4	2	3	3	1.667

Table 4.10.: Results of Age Sanity Testing on self-generated inputs and their averaged MAEM over all runs.

Conclusion

Interestingly, age identification works to some extent, especially when looking at the MAEM. Moreover, the sex does not seem to have any influence on the quality of the classification. However, the setting seems to matter. This makes it difficult to judge whether the classifier learned the classification based on voice or other factors in the recordings. It is quite conceivable that the way different age groups dictate a text differs so much that they can be distinguished.

4.5. Sex Identification

Table 4.11 shows the results on sex identification. It is noticeable that, with only 1'000 samples per sex in training, an F1-score above 0.85 is already achieved. Furthermore, the standard deviation between repeated runs is very low, except for experiment SEX-3, which had one run with an F1-score of only 0.72, resulting in a high standard deviation.

ID	volume per class	average F1	standard deviation
SEX-1	1'000	0.869	0.004
SEX-2	2'000	0.874	0.006
SEX-3	3'000	0.804	0.059
SEX-4	4'000	0.900	0.022

Table 4.11.: Results of Sex Identification experiments and their averaged F1 over all runs.

When investigating the worst run of SEX-1 and the best run of SEX-3 in Figure 4.9, an apparent reduction of the produced errors can be observed. It is around 50 % less for the *male* miss-classifications, and for the *female* classifications, the reduction is around 30 % less. This is an excellent result as the best run had only 66 miss-classifications throughout the total 1'991 evaluations, which is about 3.3 %.

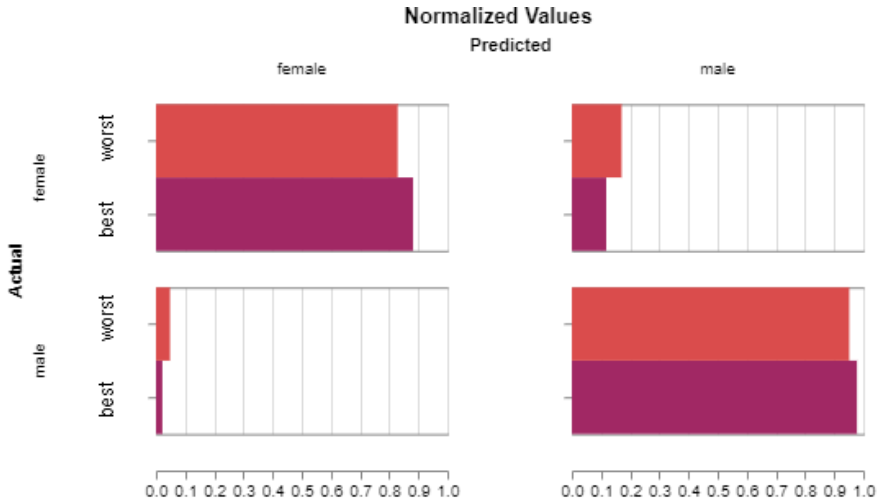
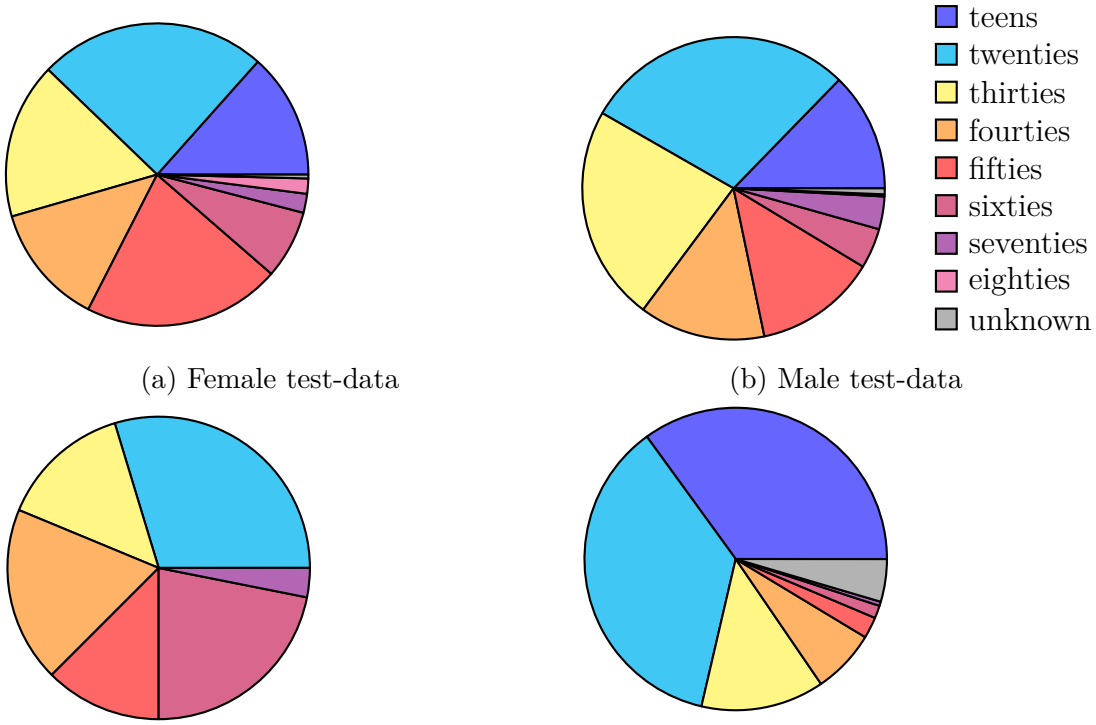


Figure 4.9.: Confusion Matrix of best SEX-4 and worst SEX-1 performing sex identification model.

Another interesting finding that emerged subsequently concerns the age distribution of miss-classifications per sex. The miss-classified samples from all three runs of the SEX-4 experiment were reviewed and grouped by sex and age. The results in Figure 4.10 show that more than two-thirds of all miss-classifications relate to *teens* and *twenties* for the *male* samples. At the same time, no specific trend can be identified for *female* miss-classifications. Although most miss-classifications for *females* also relate to *twenties*, there are none for *teens*. This was followed by examining whether there were any *female*

teens at all in the test data, which yielded 33 samples, representing 13.4 % of all *female* samples. Overall, the observed pattern seems plausible since children have high-pitched voices. This contrast is particularly pronounced in *males*, making it more difficult, even for humans, to identify a child's sex by voice alone.



(a) Female test-data

(b) Male test-data

(c) Female miss-classifications

(d) Male miss-classifications

Figure 4.10.: Distributions of test-data and miss-classifications by sex and age of SEX-4.

Since the corpus is in German, it was easy to subject the results to a more in-depth examination: It turned out that 284 errors were produced by only 66 different speakers across all repetitions of the experiment. Of these, 18 speakers produced errors in all three runs, with a total of 155 errors. Therefore, the samples of these speakers were listened to more closely. The associated findings are documented in Table 4.12.

number of speakers	total of samples	findings of analysis
9	75	young and hard to distinguish
4	41	wrong metadata
3	18	poor German pronunciation
2	21	older female
18	155	total

Table 4.12.: Findings of Sex Sanity Testing when listening to miss-classified samples.

These show that it is indeed challenging to distinguish between young people. A total of nine speakers and 75 samples fell into this category. Furthermore, it stood out that

there seems to be miss-labelled metadata in the corpus. Four speakers and 41 samples are, in our opinion, incorrectly labelled. However, it is worth mentioning that the survey in Common Voice is done under the term gender and not sex. Therefore, these four individuals may identify with this gender, making the assessment difficult in this context. In addition, three speakers were found to have poor German pronunciation and two older female speakers. Together they produced another 39 errors.

Conclusion

It is impressive how well wav2vec can determine sex. Especially when looking at the results of the sanity checks, which show that the model has most of the same difficulties as if the job was given to a human. It is also possible to achieve good results with only a small amount of training data. What also stands out is that the same speaker is classified chiefly the same way, regardless of what he says. This shows in this context the importance of having speaker independence, especially in evaluation, as it can quickly distort an F1-score otherwise.

5. Discussion and Outlook

Wav2vec is a powerful tool that currently achieves the best scores in ASR with little data. It learns speech representation based on which speech classification systems can be trained. Through experiments in the areas of AID, age identification and sex identification, this thesis has demonstrated further possibilities. Many insights that can contribute to training successful speech classifiers based on wav2vec have been gained in this process.

When AID and DID are compared, it is essential to respect the difference. An accent, by definition, describes a different pronunciation, while a dialect is a separate form of language with its own grammar and words besides different pronunciation. Therefore AID is a more complex task than DID. Related to this, it was found that it depends on how the data of a corpus is collected. In this sense, a distinction was made between freely-spoken speech, which corresponds more to the setting of DID and read-speech, which corresponds more to the setting of AID. Therefore, the results presented in this thesis can not compete with the achievements of DID in Arabic. All experiments conducted were repeated three times. Therefore, the observed differences cannot be considered statistically significant. For that reason, the results should be treated with a certain degree of caution. However, as the experiments have been conducted on low-resource datasets, our conclusions focus on these amounts explicitly.

The experiments in AID have shown that it is possible to train a classifier on wav2vec to distinguish between accents with only around eight hours per accent. It turned out to be difficult to distinguish similar accents. In evaluations with grouped accents, however, better results have been achieved. In addition, a trained classifier was able to detect wrong labelled samples in the test-data as shown with the Indian accent. The SLE experiment has shown that shorter samples are more challenging to classify than longer samples. It is unclear whether longer samples or shorter samples should be used for training to improve the results. Both seem to have potential because in training with shorter samples, the training would be better adapted to the evaluation, whereas in training with longer samples, more interconnected features of an accent can be learned. Most Voting could not cause an increase in accuracy of the prediction of short samples. As the original samples were already short, only a small number of votes could be given. Several accents often received the same number of votes, which added certain randomness from which it was not possible to benefit. However, age and sex classification have proved that wav2vec's ability in learning speech representation can be used for entirely unrelated ASR tasks. However, the age classification experiment has experienced a dependency on the audio recording setting. Therefore, it cannot be ruled out that the classification is based on factors other than speech. In contrast to the AID experiment, the sex classification reached high scores with a low-resource dataset.

This thesis has succeeded in showing that wav2vec can be used for a wide range of applications in speech classification as it was possible to classify speech with a small amount of data. Moreover, even if the scores obtained do not yet correspond to the desired values, the insights gained can be used to explore the field further.

Training neural networks is a computationally expensive task, especially when building on top of large pre-trained deep neural networks as wav2vec XLSR. Through limitations in time and computing capacity, it was not possible to conduct further thorough investigations. We propose to investigate whether it makes sense to focus on grouping similar dialects. Therefore, better results could be achieved if little data is available. On the other hand, it is equally interesting how wav2vec would behave with much more data. The data could be more valuable if it contained freely-spoken speech to represent a more realistic scenario for DID. Also, concerning training, better results could be achieved with longer samples. In terms of most voted classification, it could be possible that there are opportunities in evaluating longer audio samples. As seen in the age and sex experiment, speakers are consistently classified into the same class. Therefore it could be possible that speaker identification is another scenario where wav2vec's characteristics could be applied.

Bibliography

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervisedlearning of speech representations", Facebook AI, Tech. Rep., Oct. 2020.

[2] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification", Institute of Automation, Chinese Academy of Sciences, China and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, Tech. Rep., Jan. 2021.

[3] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition", Columbia University, Tech. Rep., 2011.

[4] A. Lazaridis, E. Khoury, J.-P. Goldman, M. Avanzi, S. Marcel, and P. N. Garner, "Swiss french regional accent identification", in *Odyssey 2014*, 2014, pp. 106–111.

[5] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language", in *Interspeech 2016*, 2016, pp. 2001–2005. DOI: 10.21437/Interspeech.2016-129. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-129.

[6] A. Abad, E. Ribeiro, F. Kepler, R. Astudillo, and I. Trancoso, "Exploiting phone log-likelihood ratio features for the detection of the native language of non-native english speakers", in *Interspeech 2016*, 2016, pp. 2413–2417. DOI: 10.21437/Interspeech.2016-1491. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1491.

[7] F. Weninger, Y. Sun, J. Park, D. Willett, and P. Zhan, "Deep learning based mandarin accent identification for accent robust asr", Nuance Communications, Inc., Burlington, MA, USA and Nuance Communications, Inc., Aachen, Germany, Tech. Rep., Sep. 2019.

[8] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic mgb-3", 1Qatar Computing Research Institute, HBKU, Doha, Qatar, Centre for Speech Technology Research, University of Edinburgh, UK, Tech. Rep., Sep. 2017, p. 7.

[9] A. A. Najim, D. P. Cardinal, S. Khurana, S. H. Yella, J. Glass, P. Bell, and S. Renals, "Automatic dialect detection in arabic broadcast speech", Qatar Computing Research Institute, HBKU, Doha, Qatar; MIT Computer Science, Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA; JHU Center for Language, and Speech Processing (CLSP), Baltimore, MD, USA; Ecole de technologie sup´erieure, D´epartement de G´enie L ´ ogiciel et des TI, Montr´eal, Canada; Centre for Speech Technology Research, University of Edinburgh, UK, Tech. Rep., Aug. 2016, p. 1.

[10] S. Shon, A. Ali, and J. R. Glass, "Convolutional neural networks and language embeddings for end-to-end dialect recognition", *CoRR*, vol. abs/1803.04567, 2018. arXiv: 1803.04567. [Online]. Available: http://arxiv.org/abs/1803.04567.

[11] P. Parikh, K. Velhal, S. Potdar, A. Sikligar, and R. Karani, "English language accent classification and conversion using machine learning", Department of Computer Engineering, DJSCE, Mumbai 400056, India, Department of Computer Engineering, DJSCE, Mumbai 400056, India, Tech. Rep., May 2020, p. 5.

[12] D. Honnavalli and S. Shylaja, "Supervised machine learning model for accent recognition in english speech using sequential mfcc features", in *Advances in Artificial Intelligence and Data Engineering*, N. N. Chiplunkar and T. Fukao, Eds., Singapore: Springer Singapore, 2021, pp. 55–66.

[13] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition", in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds., ISCA, 2019, pp. 3465–3469.

[14] A. Baevski, S. Schneider, and M. Auli, "Vq-wav2vec: Self-supervised learning of discrete speech representations", *CoRR*, vol. abs/1910.05453, Oct. 2019.

[15] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition", Facebook AI, Tech. Rep., Dec. 2020.

[16] C. Dictionary. (May 29, 2021). "Dialect", [Online]. Available: https://dictionary.cambridge.org/de/worterbuch/englisch/dialect. [29.05.2021].

[17] C. Dictionary. (May 29, 2021). "Accent", [Online]. Available: https://dictionary.cambridge.org/de/worterbuch/englisch/accent. [29.05.2021].

[18] O. for National Statistics UK. (Feb. 21, 2019). "What is the difference between sex and gender?", [Online]. Available: https://www.ons.gov.uk/economy/environmentalaccounts/articles/whatisthedifferencebetweensexandgender/2019-02-21. [07.06.2021].

[19] B.-H. Juang and L. Rabiner, "Speech recognition, automatic: History", in *Encyclopedia of Language & Linguistics (Second Edition)*, K. Brown, Ed., Second Edition, Oxford: Elsevier, 2006, pp. 806–819, ISBN: 978-0-08-044854-1.

[20] K. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits", *The Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637–642, 1952.

[21] F. Germann and M. A. Ulasik, "Evaluation of automatic speech recognition systems", ZHAW Zurich University of Applied Sciences School of Engineering, Tech. Rep., Jun. 2019, p. 69.

[22] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition", *The Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, 1983.

[23] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks", Google DeepMind, London, United Kingdom, Department of Computer Science, University of Toronto, Canada, Tech. Rep., 2014, p. 9.

[24] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models", Google Brain Google Inc. Mountain View, CA 94043, USA, Tech. Rep., 2016.

[25] T. Likhomanenko, G. Synnaeve, A. Hannun, R. Collobert, and M. Auli. (Sep. 2019). "Self-supervision and building more robust speech recognition systems", [Online]. Available: `https://ai.facebook.com/blog/self-supervision-and-building-more-robust-speech-recognition-systems/`. [02.06.2021].

[26] D. M. Eberhard, G. F. Simons, C. D. Fennig, and (eds.) (2021). "Ethnologue: Languages of the world. twenty-fourth edition.", [Online]. Available: `https://www.ethnologue.com/`. [01.06.2021].

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

[28] P. Joshi. (Jun. 19, 2019). "How do transformers work in nlp? a guide to the latest state-of-the-art models", [Online]. Available: `https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/?utm_source=blog&utm_medium=demystifying-bert-groundbreaking-nlp-framework`. [02.06.2021].

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

[30] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context", in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, 2019, pp. 2978–2988.

[31] M. Bates. (Jun. 2, 2021). "Phonemes", [Online]. Available: `https://www.dyslexia-reading-well.com/phonemes.html`. [02.06.2021].

[32] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation", Computer Science and Artificial Intelligence Laboratory Massachusetts Institute of Technology, Tech. Rep., Nov. 2017, p. 1.

[33] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search", IEEE Transactions on Pattern Analysis, Machine Intelligence, Institute of Electrical, and Electronics Engineers, Tech. Rep., Jan. 2011.

[34] W. Wong. (May 17, 2020). "What is gumbel-softmax?", [Online]. Available: `https://towardsdatascience.com/what-is-gumbel-softmax-7f6d9cdcb90e`. [03.06.2021].

[35] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus", 2020, pp. 4211–4215.

[36] G. Liu, Y. Lei, and J. H. L. Hansen, "Dialect identification: Impact of differences between read versus spontaneous speech", *European Signal Processing Conference (EUSIPCO), Aalborg, Denmark*, Aug. 2010.

[37] A. Hanani, M. Russell, and M. Carey, "Human and computer recognition of regional accents and ethnic groups from british english speech", *Computer Speech & Language*, vol. 27, no. 1, pp. 59–74, 2013.

[38] S. Saxena. (May 11, 2018). "Precision vs recall", [Online]. Available: `https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488`. [08.06.2021].

[39] B. Shmueli. (Jul. 3, 2019). "Multi-class metrics made simple, part ii: The f1-score", [Online]. Available: `https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1`. [03.06.2021].

[40] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter", Qatar Computing Research Institute, Hamad bin Khalifa University, Qatar Department of Computer Science and Engineering, The Ohio State University USA, IBM Watson Health Research USA, Johns Hopkins University USA, Tech. Rep., Jun. 2016, p. 7.

[41] J. Brownlee. (Nov. 18, 2016). "What is a confusion matrix in machine learning", [Online]. Available: `https://machinelearningmastery.com/confusion-matrix-machine-learning`. [15.05.2021].

[42] J. Brownlee. (Aug. 31, 2018). "How to use roc curves and precision-recall curves for classification in python", [Online]. Available: `https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python`. [15.05.2021].

[43] Mozilla. (May 19, 2021). "We're building an open source, multi-language dataset of voices that anyone can use to train speech-enabled applications", [Online]. Available: `https://commonvoice.mozilla.org/en/datasets`. [19.05.2021].

[44] Z. Alyafeai. (Apr. 2, 2021). "Klaam", [Online]. Available: `https://github.com/ARBML/klaam`. [09.04.2021].

[45] Z. Alyafeai. (Apr. 6, 2021). "Wav2vec2-large-xlsr-dialect-classification", [Online]. Available: `https://huggingface.co/Zaid/wav2vec2-large-xlsr-dialect-classification/tree/main`. [09.04.2021].

[46] W&B. (May 21, 2021). "Developer tools for machine learning", [Online]. Available: `https://wandb.ai/site`. [28.05.2021].

Glossary

Adam derived from adaptive moment estimation, is a popular optimization algorithm used in the field of deep learning. 24

andino Spanish accent spoken in the central Andes in South America. 20, 33

Automatic Speech Recognition Automatic transcription of speech.. 6, 7

caribe Spanish accent spoken on the islands in the Caribbean Sea. 20, 33

centrosurpeninsular Spanish accent spoken on the south central peninsula in Spain. 20, 33

Connectionist Temporal Classification Classification type and scoring function used in recurrent neural networks. . 10

Gumbel softmax continous distribution that approximates samples from a categorical distribution, enabling backpropagation.. 13

Hugging Face Framework for building, training and deploying state of the art models powered by the reference open source in natural language processing.. 22, 24

Klaam Arabic dialect classifier based on wav2vec 2.0 XLSR.. 22, 23

mexicano Spanish accent spoken in Mexico in the southern portion of North America . 20, 33

nortepeninsular Spanish accent spoken on the north peninsula in Spain. 20, 33

rioplatense Spanish accent spoken mainly around the Río de la Plata Basin of Argentina and Uruguay. 20, 33

scikit-learn Free machine learning library for Python which also provides an api for calculating metrics.. 24

scipy Python-based ecosystem of open-source software for mathematics, science, and engineering.. 28

surpeninsular Spanish accent spoken on the south peninsula in Spain. 20, 33

wav2vec 2.0 A Framework for Self-SupervisedLearning of Speech Representations created by Facebook AI. 6

List of Figures

List of Tables

Acronyms

ADI5 five classes Arabic Dialect Identification. 21–23

AID Accent Identification. 2, 7, 15, 16, 19, 20, 24, 26, 27, 35, 44

ARBML Arabic Machine Learning. 22

ASR Automatic Speech Recognition. 2, 6–8, 10, 16, 44

BERT Bidirectional Encoder Representation from Transformers. 12–14

Common Voice Mozilla's Common Voice. 2, 15, 19–21, 24, 43

CTC Connectionist Temporal Classification. 10

DID Dialect Identification. 2, 7, 15, 16, 44, 45

EGY Egyptian. 15, 21

GLF Gulf or Arabian Peninsula. 15, 21

GMM Gaussian Mixture Modelling. 7

LAV Levantine. 15, 21

LID Language Identification. 15, 16

MAE Mean Absolute Error. 17, 18

MAEM Macro Averaged Mean Absolute Error. 2, 18, 24, 27, 39, 40

MLM Masked Language Modeling. 12

MSA Modern Standard Arabic. 15, 21

NOR North African. 15, 21

PR-Curve Precision-Recall Curve. 18

ROC-Curve Receiver Operating Characteristic Curve. 18

SLE Sample Length Evaluation. 24, 44

W&B Weights & Biases. 24

wav2vec wav2vec 2.0. 2, 6, 8, 11, 13, 14, 18, 19, 22, 24, 27, 29, 35, 43–45

A. Appendix

A.1. Training details

A.1.1. English Accent Identification

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
AID-EN-1	8h 8m	0.3129	0.3757	0.3480	0.345533	0.025697
AID-EN-2	14h 18m	0.3635	0.3795	0.3445	0.362500	0.014306
AID-EN-3	20h 27m	0.3673	0.3749	0.3680	0.370067	0.003430
AID-EN-4	25h 42m	0.3861	0.3616	0.3549	0.367533	0.013411
AID-EN-5	32h 22m	0.4134	0.4081	0.3667	0.396067	0.020878

Table A.1.: English accent identification results

A.1.2. Spanish Accent Identification

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
AID-ES-1	4h 53m	0.2653	0.2426	0.2660	0.257967	0.010870
AID-ES-2	8h 59m	0.2461	0.1773	0.2531	0.225500	0.034202
AID-ES-3	13h 19m	0.2614	0.2354	0.2762	0.257667	0.016864
AID-ES-4	17h 25m	0.2889	0.2682	0.2397	0.265600	0.020170

Table A.2.: Spanish accent identification results

A.1.3. Age Identification F1

ID	time/run	F1 run 1	F1 run 2	F1 run 3	average	standard deviation
AGE-1	3h 44m	0.2915	0.3278	0.2891	0.302800	0.017705
AGE-2	7h 7m	0.3243	0.3340	0.3622	0.340167	0.016075
AGE-3	10h 34m	0.3498	0.3560	0.3475	0.351100	0.003590
AGE-4	14h 40m	0.3428	0.3441	0.3932	0.360033	0.023458

Table A.3.: Age identification results F1

A.1.4. Age Identification MAEM

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
AGE-1	3h 44m	1.072	1.053	1.105	1.076667	0.021484
AGE-2	7h 7m	1.004	1.034	0.9756	1.004533	0.023845
AGE-3	10h 34m	1.01	0.9988	1.009	1.005933	0.005061
AGE-4	14h 40m	1.01	1.009	0.9259	0.981633	0.039412

Table A.4.: Age identification results MAEM

A.1.5. Sex Identification

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
SEX-1	1h 36m	0.8638	0.8734	0.8711	0.869433	0.004093
SEX-2	2h 48m	0.8758	0.8813	0.8661	0.874400	0.006284
SEX-3	4h 1m	0.7192	0.8507	0.8375	0.802467	0.059125
SEX-4	5h 12m	0.8712	0.9235	0.9061	0.900267	0.021746

Table A.5.: Sex identification results

A.2. Evaluation

A.2.1. Spanish Binary Evaluation

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
SL-ES-4	4m 48s	0.7610	0.6648	0.7111	0.712300	0.039283

Table A.6.: Spanish sample length evaluation results

A.2.2. English Sample length

ID	length	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
SL-EN-1	1s	38m 57s	0.1520	0.1817	0.1634	0.165700	0.012234
SL-EN-2	2s	25m 41s	0.2471	0.2556	0.2216	0.241433	0.014447
SL-EN-3	3s	20m 20s	0.2889	0.2916	0.2560	0.278833	0.016183
SL-EN-4	4s	19m 5s	0.3159	0.3146	0.2722	0.300900	0.020301
SL-EN-5	5s	16m 4s	0.3334	0.3165	0.2962	0.315367	0.015208

Table A.7.: English sample length evaluation results

A.2.3. Spanish Sample length

ID	length	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
SL-ES-1	1s	22m 25s	0.1634	0.1288	0.1427	0.144967	0.014216
SL-ES-2	2s	12m 13s	0.2058	0.1571	0.1861	0.183000	0.020002
SL-ES-3	3s	8m 49s	0.2272	0.1978	0.1957	0.206900	0.014380
SL-ES-4	4s	7m 24s	0.2388	0.2148	0.2110	0.221533	0.012308
SL-ES-5	5s	6m 30s	0.2441	0.2232	0.2071	0.224800	0.015147

Table A.8.: Spanish sample length evaluation results

A.2.4. English most voted

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
MV-EN-1	21m 7s	0.0984	0.0626	0.1353	0.098767	0.029681
MV-EN-2	17m 49s	0.2504	0.1930	0.2209	0.221433	0.023436
MV-EN-3	17m 21s	0.2836	0.2757	0.2435	0.267600	0.017344

Table A.9.: English most voting evaluation results

A.2.5. Spanish most voted

ID	time/run	f1 run 1	f1 run 2	f1 run 3	average	standard deviation
MV-ES-1	22m 59s	0.1631	0.1343	0.1742	0.157167	0.016823
MV-ES-2	19m 46s	0.2159	0.1753	0.2161	0.202413	0.019152
MV-ES-3	19m 28s	0.2324	0.1954	0.2151	0.214306	0.015123

Table A.10.: Spanish most voting evaluation results

A.2.6. Age Sanity Tests

sex	run 1	run 2	run 3	sum	percentage	corpus	percentage
female	155	150	152	457	12.58 %	245	12.15 %
male	1'033	1'054	1'009	3'096	85.24 %	1'734	86.01 %
other	14	13	10	37	1.02 %	19	0.94 %
unknown	12	17	13	42	1.16 %	18	0.89 %
total	1'214	1'234	1'184	3'632	100 %	2'016	100 %

Table A.11.: Age Corpus and miss-classifications distribution by sex

condition	AGE-4 run 1 twenties	AGE-4 run 1 fifties	AGE-4 run 2 twenties	AGE-4 run 2 fifties	AGE-4 run 3 twenties	AGE-4 run 3 fifties	avg. twenties	avg. fifties	macro avg.
normal	0.5	2	0	2	0.25	2	0.250	2.000	1.125
noise	0.75	3.5	0.25	1.5	0.75	1.5	0.583	2.167	1.375
far from mic	0.75	3.5	0	2.5	0.5	2	0.417	2.667	1.542
close to mic	0.75	2.5	0	3.5	0.75	2.5	0.500	2.833	1.667

Table A.12.: MAEM of age sanity tests per class

A.2.7. Sex miss-classifications

age	SEX-4 run 1 female	SEX-4 run 1 male	SEX-4 run 2 female	SEX-4 run 2 male	SEX-4 run 3 female	SEX-4 run 3 male	total female	total male
teens	0	38	0	17	0	22	0	77
twenties	5	38	8	16	6	26	19	80
thirties	0	20	4	2	5	7	9	29
fourties	4	11	4	1	4	3	12	15
fifties	0	4	4	0	4	1	8	5
sixties	1	2	7	0	6	1	14	3
seventies	0	1	2	0	0	0	2	1
eighties	0	0	0	0	0	0	0	0
nineties	0	0	0	0	0	0	0	0
unknown	0	8	0	1	0	1	0	10
total	10	122	29	37	25	61	64	220

Table A.13.: Absolute number of sex miss-classifications grouped by age

age	total female	percentage	total male	percentage	overall percentage	macro averaged
teens	0	0.00 %	77	35.00 %	27.11 %	17.50 %
twenties	19	29.69 %	80	36.36 %	34.86 %	33.03 %
thirties	9	14.06 %	29	13.18 %	13.38 %	13.62 %
fourties	12	18.75 %	15	6.82 %	9.51 %	12.78 %
fifties	8	12.50 %	5	2.27 %	4.58 %	7.39 %
sixties	14	21.88 %	3	1.36 %	5.99 %	11.62 %
seventies	2	3.13 %	1	0.45 %	1.06 %	1.79 %
eighties	0	0.00 %	0	0.00 %	0.00 %	0.00 %
nineties	0	0.00 %	0	0.00 %	0.00 %	0.00 %
unknown	0	0.00 %	10	4.55 %	3.52 %	2.27 %
total	64	100.00 %	220	100.00 %	100.00 %	100.00 %

Table A.14.: Distribution of sex miss-classifications by age

A.3. Significance tests results

Pair		T statistic	p value	significantly different (<=0.05)
AID-EN-5	AID-EN-1	1.9978	0.1838	False
	AID-EN-2	4.0089	0.057	False
	AID-EN-3	1.8376	0.2075	False
	AID-EN-4	2.8431	0.1047	False
AID-ES-4	AID-ES-1	0.4496	0.697	False
	AID-ES-2	1.3305	0.3148	False
	AID-ES-3	0.3562	0.7557	False
SL-EN-5	SL-EN-1	9.4265	0.0111	True
	SL-EN-2	10.0728	0.0097	True
	SL-EN-3	6.1425	0.0255	True
	SL-EN-4	2.2061	0.1581	False
SL-ES-5	SL-ES-1	9.2068	0.0116	True
	SL-ES-2	3.182	0.0862	False
	SL-ES-3	4.3956	0.0481	True
	SL-ES-4	0.8845	0.4698	False
MV-EN-3	MV-EN-1	5.3823	0.0328	True
	MV-EN-2	2.4926	0.1302	False
MV-EN-1	SL-EN-1	-2.4697	0.1322	False
MV-EN-2	SL-EN-2	-0.9376	0.4474	False
MV-EN-3	SL-EN-3	-3.5949	0.0694	False
MV-ES-3	MV-ES-1	6.7664	0.0212	True
	MV-ES-2	1.821	0.2102	False
MV-ES-1	SL-ES-1	1.2511	0.3374	False
MV-ES-2	SL-ES-2	3.3635	0.0782	False
MV-ES-3	SL-ES-3	1.1583	0.3664	False
AGE-4	AGE-1	2.2428	0.1541	False
	AGE-2	3.2719	0.0821	False
	AGE-3	0.4845	0.6759	False
SEX-4	SEX-1	2.4664	0.1325	False
	SEX-2	1.6966	0.2319	False
	SEX-3	3.6052	0.0691	False

Table A.15.: F1 Significance test results

Pair		T statistic	p value	significantly different (<=0.05)
AGE-4	AGE-1	-2.2438	0.154	False
	AGE-2	-1.4212	0.2912	False
	AGE-3	-0.8224	0.4973	False

Table A.16.: MAEM Significance test results

A.4. Code

The code used for this thesis can be viewed on Github.com :
`https://github.com/DReiser7/w2v_did`