POLITECNICO DI TORINO

Corso di Laurea Magistrale in Data Science And Engineering

Tesi di Laurea Magistrale in

Exploring Incrementality in Data Subgroups for Speech Models

RELATRICE	**CORELATORE**	**CANDIDATO**
Prof.ssa Eliana Pastor	Alkis Koudounas	Leonardo Moraglia

[...] And perhaps that's the reason that we fascinate you so, because our puny behaviour shows you a glimmer of the one thing that evades your omnipotence: a moral center. And if so, I cannot think of a crueller irony than destroying us, whose only crime is to be too human.

Abstract

This dissertation addresses the analysis of performance disparities between subgroups of data within speech recognition models. More specifically, a particular focus is put on how the performance of subgroups evolves during the training of the models themselves.
Having considered previous studies that have highlighted the presence of subgroups and discriminatory biases in speech recognition models, this work also focus on understanding how these disparities are created and propagated during training. For instance, populations characterized by attributes such as gender, accent, speech rate, or age may undergo either a decline or enhancement in the model's performance as it develops.
In conclusion, we conducted a detailed examination of the evolution of these disparities across diverse datasets and speech recognition models to acquire an in-depth understanding of their propagation and found that although overall accuracy improved across different models, some subgroups continue to exhibit notable performance gaps. This analysis aims to contribute to the advancement of these technologies towards enhanced fairness and accessibility in the future.

Contents

Chapter 1

Introduction

The expanding deployment of speech models in applications—from virtual assistants to accessibility technologies—necessitates uniform performance across diverse demographic and linguistic groups. Although the advancement of speech recognition systems has led to significant improvements in overall accuracy, noticeable discrepancies still persist in model performance across population subgroups. This being, for example, attributes such as gender, accent, age, emotional expression, and pitch during the dialogue with a vocal assistant. These discrepancies not only impact the overall model accuracy, but are real challenges to the inclusivity and fairness of these technologies.
As Koenecke et al.'s study pointed out, we see that in the most currently used ASR speech recognition systems, the African American Vernacular English (AAVE) speakers tend to have higher Word Error Rates (WER) when compared to speakers of standard American English. The fact that the percentage of incorrectly predicted words is higher for the African American subgroup points to language barriers that are well-present in many environments, showing how disparities between subgroups can be perpetuated even in extremely well-established speech recognition models.
Building on the systematic approach established in *"Towards Comprehensive Subgroup Performance Analysis in Speech Models"*[1], this thesis adopts a structured methodology to gain a deeper understanding of subgroup-specific performance trends in speech models and therefore helps us to comprehend how Spoken Language Understanding (SLU) models perform across distinct user demographics, revealing biases and performance discrepancies that may otherwise remain obscured.
With the examination of subgroup performance under controlled conditions, this study aims to underscore the necessity for dedicated evaluation metrics that precisely capture model behaviour in real-world scenarios, where diverse user characteristics are notably prevalent.

During the development of this thesis, there will be a detailed discussion on how the performance of a range of subgroups varies throughout the training of machine learning models for Spoken Language Understanding.
To gain this insight on how disparities evolve and propagate during the training of different models, this study will train multiple models using different datasets, allowing us to systematically analyse the variations in model performance and bias emergence, thereby gaining a deeper understanding of their underlying mechanisms. With this analysis, it will be possible to observe how certain disparities, such as gender or speaking rate, have an impact on the performance of the model, both positively and negatively.

This dissertation also makes use of techniques and the research "Prioritising Data Acquisition for End-to-End Speech Models" [2] and "Exploring Subgroup Performance in End-to-End Speech Models" [3], such as methodologies on subgroup-specific error analysis and confidence-based evaluation. In addition to these, this dissertation will make use of DivExplorer, a tool developed by E.Pastor in *"Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence"*[4] to perform subgroup-identification strategies and assisting in the identification of underperforming subgroups and subgroup performance analysis techniques to assess subgroup disparities during the models training.

Through this research, we will attempt to understand the manner in which subgroup disparities might occur during model training and how it might introduce performance disparities through subgroups, potentially contributing to systematic bias and unfair treatment. Lastly, one might take the results presented in this dissertation as a baseline for the study and implementation of adaptive training techniques dynamically adjusting training weights to mitigate subgroup disparities, fostering more equitable model performance, and enhancing overall model quality.

1.1 Motivation

1.1.1 Uncovering the Temporal Dynamics of Subgroup Disparities

Understanding how subgroup disparities are propagated throughout the model training cycle is essential. Not only does this enable an understanding of

how these disparities propagate, but it also offers insight into the model's evolution during training, allowing for real-time adjustments to improve its subgroup performance. That is exactly why the main motivation for this study is to gain a deeper understanding of the propagation of subgroup disparities in models specialized in spoken language understanding.

We might, for instance, have a situation where a basic model behaves and performs quite well for all types of speech, but once trained on a new dataset, the resulting model might end up having a higher precision and accuracy over slow male speech while it may have an harder time understanding a faster female voice or one with a different accent. The introduction of these disparities could also happen if, during one of the model training steps, a subgroup consisting of slow-speaking women performs better than fast-speaking men. Herein lies the motivation behind this study, the desire to gain a greater understanding of how these disparities propagate through model training.

These disparities, with varying degrees of severity, are inherently present despite the outcome of the final model [5]. This is because standard machine learning models, despite their complexity, seek to achieve the highest percentage of accuracy while very often overlooking or assigning lower significance to aspects that do not directly contribute to optimal overall performance.

Recent studies, such as that of Dheram et al. (2022) [6], remind us that there are certain turning points during the training of models at which the performance of subgroups undergoes sudden changes. This research proposes to find these inflection points and study them in order to gain a deeper understanding of the architectural factors of the models that determine these changes, in the hope of paving the way for adaptive and dynamic training procedures that can re-weight themselves in response to disparities and the presence of subgroups.

1.1.2 Guiding Fairness-Driven Interventions and Adaptive Training

The second motivation behind this dissertation is the intention to understand how subgroups incrementally propagate within models during their training. The objective is to identify ways to intervene more proactively in the training in order to achieve a more generalised fairness of the model, with better effectiveness than post hoc adjustments. In fact, the latter, in addition to

being later applied, merely apply to the disparities that emerge from the final model, which might omit others that are perhaps minor or eluded the analysis. Conventional approaches also consider a change of weights or perhaps a downsizing of the classes in the training datasets, which, only after training the model, will supposedly create fewer disparities.
Therefore, we wish to investigate the area of real-time disparity tracking during the training of the model. This would allow us to intervene on a step-by-step basis by correcting and modifying the model's internal weights in such a way as to intervene on disparities without costly post-training intervention.
Returning to the example cited earlier, if we observe during a training phase that the model performs better on individuals who speak slowly to the detriment of those who speak faster, it might be possible to calibrate the model automatically via adaptive weighting methodologies to correct for these disparities as they arise. In addition to correcting disparities, this approach would also make the model better able to tackle instances of subgroup stagnation when it occurs. As seen in the study by Koudounas et al. (2024) [2], it is demonstrated that once performance disparities are identified, one can use them to build data acquisition strategies to address them. More specifically, their approach prioritizes the collection of samples from subgroups exhibiting lower performance, thereby addressing weaknesses in model training. By adopting a divergence-aware acquisition strategy, they show that targeted data selection can improve both overall model accuracy and fairness across subgroups, making it a viable complement to real-time disparity tracking.

Similarly, Dheram et al. (2022) [6] show how early intervention can reduce discrimination. Indeed, through over-sampling of the lower-performing subgroups resulted in improving the model's WER (Word Error Rate) between the best-performing and the lowest-performing subgroups from 56% to 38%.

We then consider that the limited existing research focuses solely and exclusively on evaluating the performance of subgroups of models that have already finished their entire training cycle. In this study, in fact, we want to propose a more granular and incremental approach by examining the performance of subgroups during the training process.

1.2 Problem

This thesis aims to address the problem of subgroup disparities arising during the training of speech recognition models. As has already been shown in the previously mentioned studies [3, 1, 6, 7], the already trained models are prone to the existence of disparities in speech comprehension for different subgroups. This thesis aims to address this problem in greater depth, namely to understand how these disparities in speech recognition can be introduced and progress during the training of models. Such distortions will eventually lead the model to perform better for some subgroups at the expense of others. Ultimately, it is essential to ensure that individuals with different accents and speech patterns are not effectively "discriminated against" in situations where their voice is not understood.

Smart home devices such as Google Home and Amazon Alexa are prone to misunderstanding or directly misinterpreting what the user is saying [8] just because they have spoken with a particular accent or their speech has not been optimized in training. We recognise that these discrepancies may be introduced by a variety of factors, an imbalance of subgroups in the training dataset, with perhaps the presence of one accent at the expense of others. Otherwise, perhaps the problem could be found in the structure of the model itself, which, in its quest to minimise error, inevitably favours one speech over the others.

The real-world implications of these disparities are by no means to be underestimated; it is not just a matter of inconvenience for the user, but one can easily understand how these issues can easily imply real discrimination. More recently in the USA, gestures such as the waiter taking orders are already being replaced by voice recognition models, especially in fast-food chains [9]. As technology progresses, these pioneering developments show us just how pervasive these technologies will become. If speech recognition models are not developed to deal with subgroup disparities, we may soon have cases where sub-groups of people, perhaps already poorly representative of the majority, will be further discriminated against, further perpetuating societal problems.

It is for this very reason that it is emphasised that the main issue in this thesis is that of the discrimination that speech recognition models intrinsically have and how this is to be considered. Tackling this issue is essential in order to ensure that speech recognition systems are reliable, non-discriminatory, and can bring real benefit to all users.

1.3 Objectives

The purpose of this thesis is to study the identification and understanding of subgroup disparities in speech recognition systems. More specifically, to examine their appearance and progression during incremental training on different models and datasets.

Indeed, the aim is to shed light on how these disparities, after appearing in the training of the model, propagate and influence it during its development. By concentrating on the training phase, this dissertation aims to study and investigate the mechanisms by which subgroup disparities are created, amplified, or potentially mitigated, and finally propagated until they appear in the final model.

Such analysis is founded on research previously carried out by Pastor et al. [4], which has already examined subgroup disparities across different models and datasets. The intention behind this study is to continue their work and analyse—not the presence of performance disparities of subgroups in the final model—but rather to investigate how these phenomena shape the model throughout its training phase.

It is precisely for this reason that we wish to understand how these subgroup disparities are structural causes of disparities for the model, in the desire to achieve a more theoretical as well as practical understanding of a phenomenon that remains little explored to this day, especially in speech recognition models.

With this greater understanding, the intention is to pave the way for training methodologies that actively reduce disparities, promoting more equitable speech recognition models for disparate audiences.

Chapter 2

Literature Review and State of the Art

2.1 Spoken Language Understanding (SLU)

Spoken Language Understanding (SLU) represents a core component in the development of intelligent speech systems, enabling machines to process and interpret human speech for various downstream applications, such as virtual assistants, transcription services, and accessibility tools [10, 11, 12]. SLU models are typically built on deep learning architectures that transform raw audio signals into structured representations—such as semantic slots and intents—thereby capturing the meaning behind spoken utterances. Significant advancements in SLU research have leveraged large-scale data, deep learning architectures, and sophisticated training methodologies, yet the field continues to grapple with challenges such as subgroup disparities, fairness, and bias mitigation [13].

Recent theoretical developments in deep learning further delve into these challenges. A recent article in *Quanta Magazine* from 2024 [14] discusses a novel link between modern deep neural networks and older kernel methods. They mention that idealized conditions, infinitely wide neural networks, have the exact behaviour as kernel machines. This changes the general perspective and helps to demystify why over-parameterized models can reach a good capacity for generalization despite their complexity. Even if this theoretical framework is not directly aimed at subgroup analysis, as at the time of writing this the current state of the art of SLU, it provides valuable insights into the learning dynamics that may underlie the emergence and stabilization of subgroup disparities during training.

2.2 SLU and Subgroups in Speech Models

Recent studies have highlighted the growing importance of subgroup performance analysis within SLU systems. For example, Koudounas et al. (*Assessing Speech Model Performance: A Subgroup Perspective*) emphasize the role of metadata, such as speaker demographics and acoustic features, in identifying performance disparities across subgroups. Their work demonstrates how task-, model-, and dataset-agnostic frameworks can uncover intra- and cross-model gaps, providing actionable insights for data acquisition strategies aimed at reducing these disparities.

One critical challenge is the persistence of demographic biases in SLU systems, as demonstrated by Koenecke et al. (*Racial Disparities in Automated Speech Recognition*). This study exposed significant word error rate (WER) differences between racial groups in commercial ASR systems, tracing these disparities to insufficiently diverse training datasets. Similarly, Toussaint and Ding (*SVEva Fair: A Framework for Evaluating Fairness in Speaker Verification*) propose fairness evaluation frameworks tailored to speaker verification systems, revealing consistent underperformance for female speakers and certain nationalities, underscoring the broader implications of demographic bias in SLU models.

Bias mitigation strategies have also been explored extensively. Koudounas et al. in (*A Contrastive Learning Approach to Mitigate Bias in Speech Models*) [15] introduced the CLUES framework, which leverages contrastive learning to enhance subgroup-level representations, thereby reducing performance disparities. Their approach demonstrated significant reductions in subgroup performance gaps across diverse datasets and languages, highlighting the potential of targeted representation learning techniques.

Additionally, methods focusing on post-training evaluation and correction have gained traction. For example, Baldini et al. in (*Your Fairness May Vary: Pre-trained Language Model Fairness in Toxic Text Classification*) [16] revealed the variability of fairness measures in fine-tuned language models and proposed post-processing techniques to improve fairness without retraining. These findings are relevant for SLU, where dynamic and scalable mitigation strategies are essential to addressing evolving biases.

The broader implications of fairness and inclusivity in SLU extend to practical applications. Automated systems must reliably serve diverse populations to fulfill their potential as equitable tools. Koudounas et al.[1] em-

phasize the importance of metadata-driven subgroup analyses for identifying nuanced performance discrepancies, facilitating interventions during training that improve model fairness.

2.3 Subgroup Analysis in Speech Models

The analysis of subgroup performance in speech models has emerged as a critical area of research, highlighting disparities that affect the fairness and inclusivity of these systems. Subgroup analysis focuses on understanding how different segments of a population, characterized by demographic or acoustic attributes, experience varying levels of model performance. Recent studies provide a deeper understanding of these disparities and propose methodologies to identify, analyze, and mitigate them effectively.

A key contribution in this area is the study *"Assessing Speech Model Performance: A Subgroup Perspective"* by Koudounas et al. [17], which emphasizes the importance of enriched metadata for subgroup analysis. The study incorporates speaker demographics (e.g., gender and age) and signal-related attributes (e.g., speaking rate and pauses) to uncover intra-model and cross-model performance gaps. By identifying interpretable subgroups, the methodology revealed significant disparities, such as poorer performance for specific age groups or accents, providing a framework for targeted data acquisition to improve subgroup performance.

Another influential work, *"Houston We Have a Divergence: A Subgroup Performance Analysis of ASR Models"* by Koudounas and Giobergia [18], explored subgroup disparities in automatic speech recognition (ASR) models. This study utilized metadata-rich subgroups to compare performance across multilingual and monolingual ASR systems. Notably, it found that fine-tuning reduced performance divergence among subgroups, suggesting that training strategies could significantly influence subgroup equity. Moreover, the research demonstrated that larger model sizes do not uniformly improve subgroup performance, emphasizing the complexity of subgroup-specific challenges.

In *"Assessing and Mitigating Speech Model Biases via Pattern Mining"* by Koudounas et al. [19] proposed an automated approach for identifying critical subgroups through pattern mining techniques. This method identified subgroups with the largest intra- and cross-model performance gaps and introduced data acquisition strategies that effectively reduced these dispari-

ties. The study validated its approach across multiple tasks, including intent classification and emotion recognition, underscoring its adaptability and efficacy in addressing subgroup biases.

The study *"Your Fairness May Vary: Pre-trained Language Model Fairness in Toxic Text Classification"* by Baldini et al. [16] offered insights into fairness evaluation, which has parallels in subgroup analysis for speech models. The study demonstrated that fairness characteristics can vary significantly across model architectures and tasks, highlighting the need for subgroup-level evaluations during model development. It also proposed post-processing methods for bias mitigation, which can complement subgroup-specific training interventions in speech systems.

Subgroup analysis in speech models also extends to practical applications. Martins Kronis, in *"Harvesting Targeted Speech Data from Highly Expressive Found Spontaneous Speech"* [20], explored methods to isolate specific speaker data in noisy environments. This work emphasized the importance of speaker embeddings and metadata to extract meaningful subgroups, paving the way for enhanced data curation strategies.

2.4 Gaps in the Literature

Insufficient Exploration of Incremental Subgroup Performance Dynamics

This thesis addresses a novel area within AI explainability: incremental subgroup explainability, a subject that remains largely unexplored. Currently, the closest research on this topic is that of Pastor, who identified and analyzed disparities in fully trained models [4]. Pastor's work infers that subgroup-specific biases emerge as an inherent property of the training process, hinting at the possibility that a more granular, incremental analysis of subgroup explainability could reveal previously overlooked facets of model behaviour.
Furthermore, Koudounas et al. (2023) [3] demonstrated how subgroup-level performance analysis can identify performance variations across multiple metadata-defined subgroups, reinforcing the need for finer-grained monitoring during model training.
In addition, it is worth mentioning the research of Dr. Da-Wei Zhou [21], whose studies have shown how models, this time in the context of image recognition, were able to learn new classes while regressing their performance on classes already seen. This issue, which she called catastrophic forgetting,

resulted in a loss of performance, which is another characteristic tendency for all kinds of model training. She describes various strategies to mitigate this problem, such as exemplar replay and knowledge distillation.
Nevertheless, the techniques discovered neglect any subgroups and disparities that may be introduced in them but instead focus only on the overall performance of the model. Therefore, we highlight that incremental subgroup explainability can mitigate possible biases and create models that are both fairer and more effective.

Limited Representation in Training Datasets

Although automatic speech recognition (ASR) models are becoming increasingly popular, one must be aware that the datasets used for their training are very often unbalanced. In datasets for ASR in English, there are very often demographic disparities such as gender, ethnicity, accent, other spoken languages, and social characteristics.
Even if with good intentions, there have been attempts to attempt this issue by trying to create new, more balanced datasets; however, these approaches have often been unsuccessful as they have not been scalable and have in any case failed to capture the diversity and complexity that the real world models face.
As pointed out by Koeneke et al. [22], these shortcomings result in ARS models exhibiting consistent inequalities in their performance, highlighted above all by higher error rates for certain subgroups. Among these, we have African American Vernacular English (AAVE) speakers compared to white speakers.
Indeed, Koenecke shows that the main speech recognition models, including those developed by reputable companies such as Amazon, Apple, IBM, and Microsoft, exhibit an average error rate of 35% for black speakers compared to 19% for white speakers. Such critical differences make it clear that these companies, despite having the necessary resources at their disposal, are not effectively managing the phonetics of minority subgroups. It is thus suggested that larger datasets inclusive of the subgroups be created for greater balance, including more speech and language variants such as AAVE.

Lack of comparative studies on incrementality

Another significant gap is the lack of comparative studies that analyse incrementality in both speech production and reading aloud within a single

framework. While individual studies have investigated these modalities separately, a comprehensive comparison of how incrementality manifests in different contexts remains unexplored [19]. This is crucial, as it could reveal how discourse context and cognitive resources affect planning scope differently during spoken language production and reading tasks.

Next Steps for Fair and Adaptive Speech Recognition

To address these gaps, future research should focus on developing incremental evaluation frameworks that track subgroup performance during training, enabling the identification of critical moments for intervention. Scalable data balancing strategies should prioritise realistic representation of underrepresented groups, leveraging synthesis and dynamic sampling techniques. Fairness-driven training algorithms must be integrated into the training lifecycle to dynamically address biases as they develop.
Nuanced subgroup definitions incorporating socio-linguistic and acoustic factors could enhance the granularity of performance analyses.
Systematic cross-model benchmarking and the strategic use of metadata could further inform the design of bias-resilient and universally applicable speech models [23]. These advancements would contribute to the creation of inclusive and equitable spoken language technologies.

Chapter 3

Theoretical framework

3.1 Machine learning for speech

Machine learning is a branch of computer science focused on the development of algorithms that depend on a set of examples of a given phenomenon to “learn" something from them. To learn, we refer to the construction of a statistical-mathematical model based on the examples that can infer and extract information from them. If the models reach some capacity of abstraction, they can very well be used to solve practical problems inherent to the data they have been trained with.
For example, in a spoken language understanding context, the models are trained to interpret audio commands by mapping them to structured outputs such as semantic slots or intents. The development of the model will rely on a big set of annotated data to provide a variety of examples of spoken commands and their corresponding textual meaning.

3.1.1 Supervised Learning in SLU

In supervised learning for Spoken Language Understanding (SLU), the dataset is a collection of labeled examples $\{(x_i, y_i)\}_{i=1}^{N}$. Each element x_i among N is referred to as a feature vector. A feature vector is a vector where each dimension $j = 1, \ldots, D$ contains a value that represents some aspect of the example. This value is called a feature and is denoted as $x^{(j)}$. For instance, if each example x in our dataset represents an audio command, then the first feature, $x^{(1)}$, could describe the duration of the command in seconds, the second feature, $x^{(2)}$, could describe the pitch range, and $x^{(3)}$ could describe the presence of background noise, and so on. For all examples in the dataset, the feature at position j in the feature vector always contains the same type of information. This means that if $x_i^{(2)}$ contains the pitch range for some

example x_i, then $x_k^{(2)}$ will also contain the pitch range for every example x_k, $k = 1, \ldots, N$.

The label y_i can be a class from a finite set of intents $\{1, 2, \ldots, C\}$, a semantic slot value, or a more complex structure such as a tree or a graph. Unless otherwise specified, in this context, y_i typically represents either an intent or a slot value. For instance, if the examples are audio commands for a smart assistant, the intents could be {*turn_ on_ lights*, *set_ alarm*, *play_ music*}. A class, in this context, refers to the intent category to which an example belongs. For instance, an audio command *"turn on the lights in the kitchen"* could belong to the intent class *turn_ on_ lights*.

The objective of a supervised learning algorithm in SLU is to use the dataset to build a model that takes a feature vector x as input and outputs information that helps deduce the corresponding intent or slot values for that feature vector. For instance, a model trained on audio commands could take as input a feature vector describing an utterance and output the probability of the intent being *turn_ on_ lights*.

3.1.2 Advantages of Self-Supervised Pre-Training

Precisely by using self supervised learning methods, in recent years we have seen a great development of speech recognition models which do not require large amounts of labelled data. In particular, approaches such as wav2vec 2.0 or HuBERT use architectures based on Transformers to 'mask' parts of the audio signal and predict missing portions or cluster assignments.
A major advantage of the transformers lies in their ability to analyse audio sequences in parallel, as well as using systems such as self-attention, they are able to capture long range speech and sound patterns in a more efficient and scalable manner. Further advantages of this approach are the global pattern perception of audio sequences as well as the high speed of the training process.
Such an approach allows these models to generalise an understanding of language by using extremely small labelled data sets since the model is already able to recognise a large amount of audio signals.

3.1.3 Neural Networks and Deep Learning in SLU

Deep learning is a method of machine learning that independently builds (trains) general rules as an artificial neural network from example data dur-

ing the learning process. This is especially useful in the fields of machine vision and spoken language understanding in which the neural networks are trained by supervised learning to learn complex patterns leveraging the examples initially provided.

Deep learning uses a certain form of artificial neural networks *(ANNs)*, which must first be trained with sample data and then can be used for its tasks. The use of a trained ANN is called "inference". During inference, the ANN reports back an assessment of the data supplied according to the learned rules. This can be, for example, the estimation of whether an input image represents a faulty or error-free object or, in our context, the predicted letters of a spoken sentence [24].

Neurons, Layers and Connections

An ANN consists of layers of "neurons" that are linked together. In the simplest case, these are an input layer and an output layer. The input layer receives raw data, such as features extracted from audio signals, and serves as the entry point for the network neurons. The output layer, on the other hand, provides the final predictions or results of the network, such as classification labels, probabilities, or continuous values.
The links between these two layers allow the input data to be transformed and mapped to the desired outputs through a series of mathematical operations. From a mathematical and programming perspective, one can consider neurons and their links with each other as matrices in which each link matrix contains, for each value of the input matrix, a connection to the values of the result matrix. With the values of the link matrix containing the weighting of the respective connection, the weighting of the input value with the value of the logic matrix produces the respective value in the result matrix.

Deep Artificial Neural Network

The term deep learning describes the training of so-called "deep" ANNs. These networks, besides having the input and output layer, also consist of hundreds of additional "hidden" layers between the visible layers for input and output. In this structure, the resulting matrix of a hidden layer serves as the input matrix of the next layer, and so forth. Only the output matrix of the last layer will contain the result (or prediction) of the model [25].

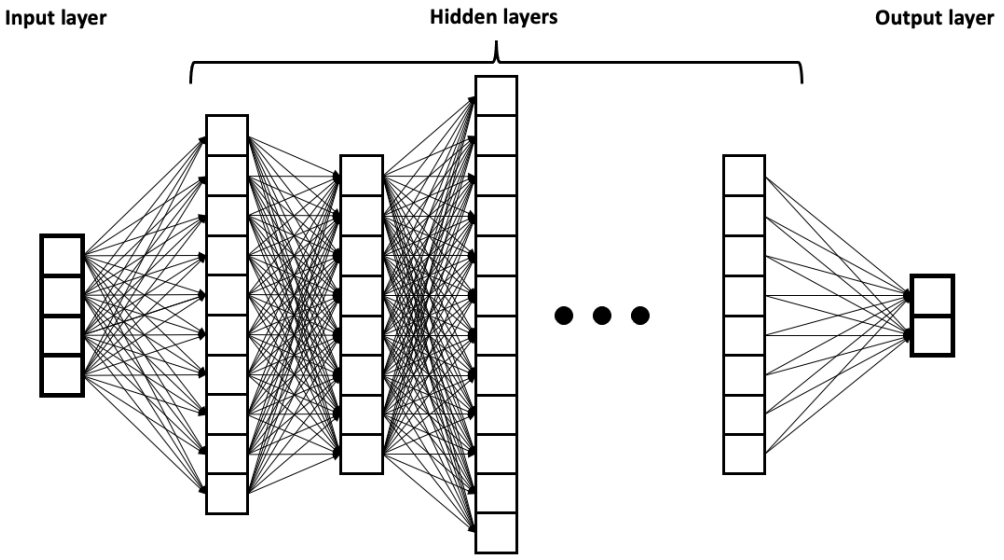

Figure 3.1: Schematic representation of a deep neural network with its information flow from the input layers to the output layer. Adapted from Resnik and Hosseini(2024) [26].

3.2 Training a Deep Artificial Neural Network

In this section, we will explore how we can train a Deep Neural Network for speech recognition.
Taking as our starting point a list of inputs and their outputs, respectively, audio files of spoken text and their own transcriptions. Our goal is to predict the output from the input. In the case of ASR, the input is an audio file and the output is its transcription; for SLU, the output is its intent.

To do so, we must then seek the relationship - or function- mapping these inputs to their output.
Within our ANN, we have several hidden layers of neurons that are nothing more than matrices whose units are connected to the previous and following layers. In each layer, the links between the neurons are represented by a matrix. This matrix holds W-weights that are initially randomly initialised and then combined with a bias factor. A weight determines the strength and direction of the connection between neurons, influencing how much the input contributes to the output, while the bias is an additional parameter that shifts the output of a neuron, adding to the model's adaptive possibilities by adjusting the activation threshold.

The training of the model sees a layer perform an operation that combines the weights of the input connections in a continuous iteration on the weights and biases. For each update, a non-linear activation function (such as ReLU or sigmoid) is applied, which introduces non-linearity into the model. This very operation makes the model, after many update steps, capable of learning complex relationships between input and output.

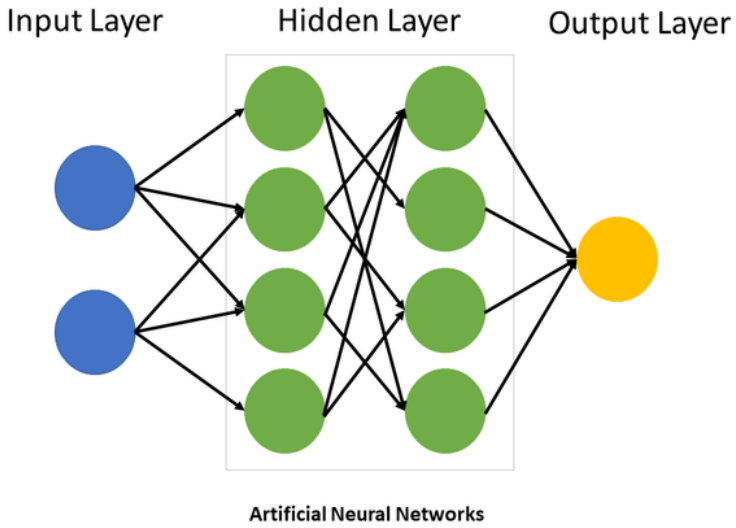

Figure 3.2: Neural network diagram**Rights:** TseKiChun, *CC BY-SA 4.0*

The aim of the model during its training is to minimise its error; this is done by means of gradient descent. Gradient descent is an optimization algorithm commonly used to train machine learning models and neural networks; its goal is to minimize errors between predicted and actual results.

A key component in this training process is the back-propagation. Back-propagation is the algorithm that efficiently computes the gradient of the loss function with respect to each weight and bias by propagating the error backward through the network. Starting from the output layer, it applies the chain rule of calculus to determine how much each parameter contributed to the error. By systematically adjusting the weights and biases according to these gradients, back propagation allows the network to learn from its mistakes and gradually improve its performance over time.

Optimizers and Gradient descent

We now come to the optimisers which are a collection of algorithms central to the training of any neural network. The main objective of these algorithms is to find the minima of a loss function in order to determine all weights and parameters of the neurons inside the neural network so that, iteration after iteration, the performance of the model can be improved both in training and in testing[25].
In our research we therefore deal with gradient descent, an optimisation algorithm that is used extremely often during the training of neural networks. The peculiarity of this algorithm is its basis on a convex function that iteratively modifies parameters with the ultimate aim of minimising a function to a minimum. This algorithm begins by pseudo-randomly defining its initial parameters and then uses the differentiation calculation to modify these values for each iteration in order to reduce the loss function and thus change the performance of the model for the better or possibly for the worse.

Being more specific, the gradient descent uses the mathematical element of the gradient. The gradient is essentially a partial derivative that is calculated with respect to the input values, this mechanism in fact allows one to go and measure the change that a weight must have based on the observed error. We can picture this algorithm graphically by imagining a straight line tangent to a curved line. The steeper the slope of the tangent line, the faster the model can learn. In cases where the slope is zero or close to zero, the model will not learn or will learn very slowly. Indeed, this slope directly influences the updates of the weights and the general bias of the model. The algorithm hence operates in a multidimensional space where it searches through these slopes to find where the loss function is most minimised, also referred to as the convergence point3.3.
Ultimately, it should be noted that in this multidimensional curvilinear space, there is an ideal minimum which, once reached, achieves an accuracy of 100%, but there are also many local minima that are more or less efficient and where the gradient descent algorithm may stop believing it has reached the global minimum. These local minima will obviously lead to a lower performance than the global minimum, but are nevertheless essential in the search for points of convergence.

To get more specific, let us now examine how the gradient is calculated mathematically. A cost function's gradient $f(\mathbf{w})$, where $\mathbf{w}$ represents the vector of parameters or weights of the model, is defined as the partial derivative vector $\nabla f(\mathbf{w})$. During each iteration, the gradient descent algorithm

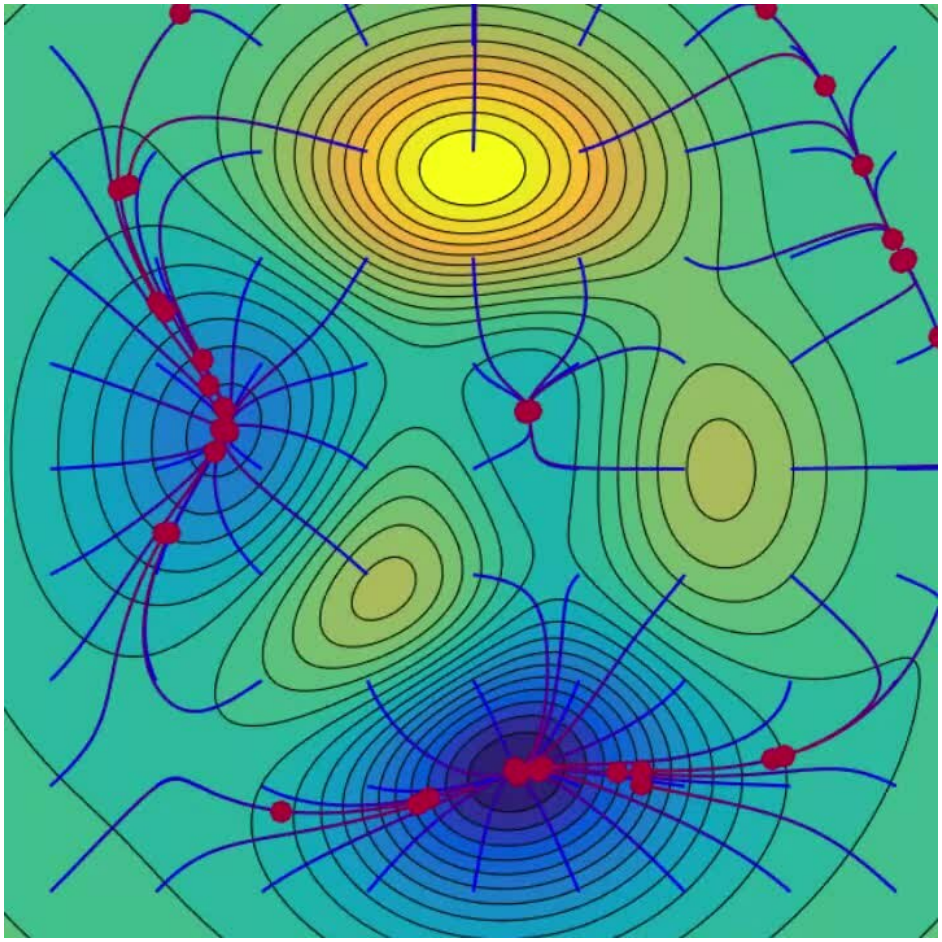

Figure 3.3: Gradient Descent in 2D. The original uploader was Gpeyre at English Wikipedia. Derivative work - This file was derived from: Gradient Descent in 2D.webm, Public Domain.

updates the weights according to the formula:

$$\mathbf{w}^{(t+1)} \;=\; \mathbf{w}^{(t)} \;-\; \eta\, \nabla f\big(\mathbf{w}^{(t)}\big),$$

in which η is the learning rate and $\nabla f\big(\mathbf{w}^{(t)}\big)$ identifies the direction of maximum growth of the cost function; therefore, subtracting this term, which is possibly scaled by η it "descends" in the direction of cost reduction.

Learning rate

The learning rate (also referred to as step size or the alpha) is the size of the steps that are taken to reach the minimum. This is typically a small value,

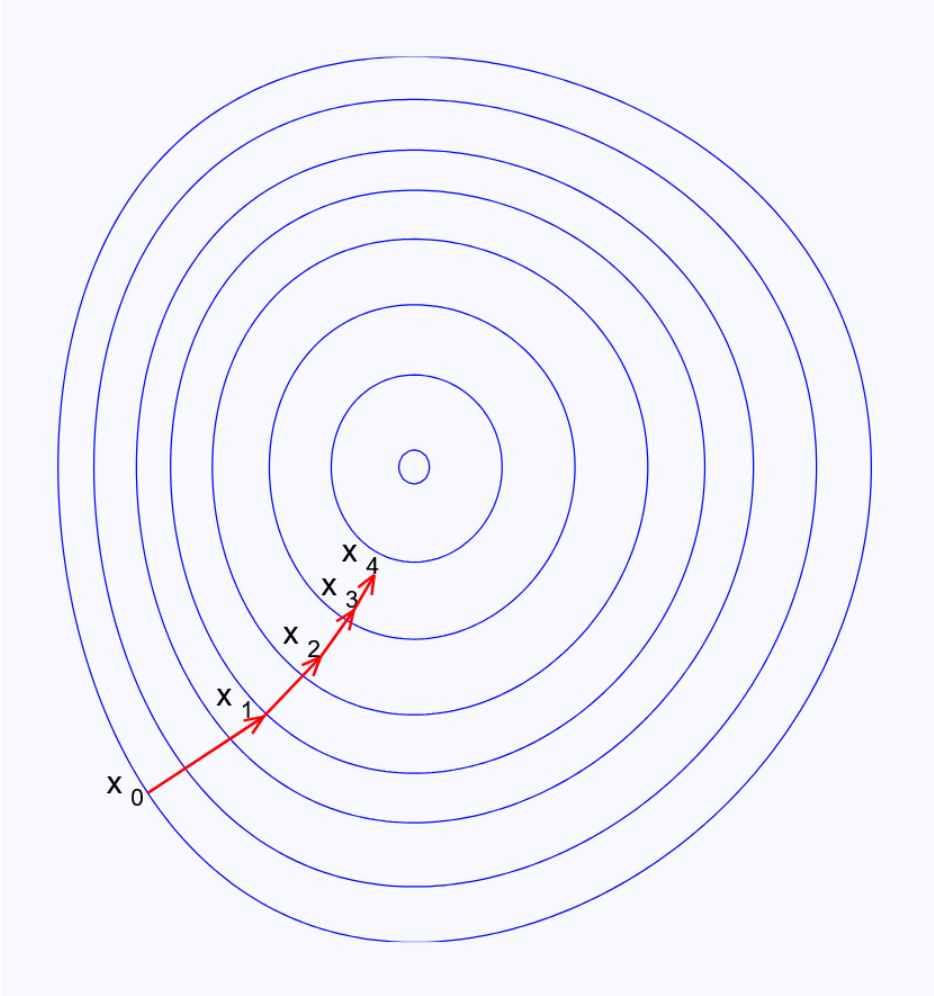

Figure 3.4: Gradient_descent.png: The original uploader was Olegalexandrov at English Wikipedia.derivative work: Zerodamage - This file was derived from: Gradient descent.png:, Public Domain

and it is evaluated and updated based on the behaviour of the cost function. High learning rates result in larger steps but risk overshooting the minimum. Conversely, a low learning rate has small step sizes.
While it has the advantage of more precision, the number of iterations compromises overall efficiency as this takes more time and computations to reach the minimum.

Cost/Loss Function

The cost (or loss) function measures the difference, or error, between actual y and predicted y at its current position. This improves the machine learning model's efficacy by providing feedback to the model so that it can adjust the parameters to minimize the error and find the local or global minimum. It continuously iterates, moving along the direction of steepest descent (or the negative gradient) until the cost function is close to or at zero. At this point, the model will stop learning. Additionally, while the terms cost function and loss function are considered synonymous, there is a slight difference between them. It's worth noting that a loss function refers to the error of one training example, while a cost function calculates the average error across an entire training set. The most common cost function is the Quadratic loss function. It's main advantage is that an error of a certain magnitude above the target produces the same loss as an error of the same magnitude below the target. If we set the target as t for some constant C, the quadratic function is:

$$\lambda(x) = C\,(t - x)^2$$

3.3 Speech Recognition

Voice and speech recognition represents one of the most fascinating areas of machine learning and artificial intelligence. The latest developments in speech recognition have, in fact, led to a simpler relationship between man and machine.
If previously, for example, people had to write, now they can dictate; if previously they had to open a browser to do a search, now they can simply activate a voice assistant with their voice. In addition to enabling more 'natural' interaction between humans and machines, this technological advancement has also become crucial for people with physical disabilities, for whom speech recognition systems have become a valuable aid towards their independence.
Over the last few years, partly due to advances in machine learning techniques and the use of deep neural networks, speech recognition models have reached very high levels of accuracy and reliability, making them easily implementable in a multitude of systems and linking them with other artificial intelligence models [27]. Of these, this thesis focuses on two in particular: automatic speech recognition and intent detection.

ASR

Automatic speech recognition, also known as ASR, is the field of speech recognition models that allows human speech to be interpreted and transcribed. Examples of ASR systems are dictators, a feature now integrated in Microsoft Word and the Google Keyboard that, when activated, can compose text using human dictation.
ASR models find, as a learning source, large datasets in which a written text is associated with its transcription. These include audio books, films, videos, and generally subtitled multimedia content, as well as text dictated and then corrected by the user.
Despite the advances of the last few years, however, it must be acknowledged that ASR systems are lacking in the understanding of different accents, dialects, and the handling of background noise [28].
However, research in ASR is progressing not only towards improving transcription capabilities but also towards semantic analysis of speech, allowing not only transcription but also the extrapolation of the intonation of sentiments and keywords in a speech stream.

Intent Detection

The detection of the intent, also called intent detection, has a fundamental role in natural language understanding. This is because it allows us, by expressing brief vocal commands, to make a machine recognise an action that we want it to perform. A simple example of intent detection can be related to a vocal assistant. If we ask it to turn on the light in the kitchen or set an alarm clock, it will follow up with understanding our command and the subsequent implementation of what we have said. To be more specific, intent detection aims at interpreting the user's communicative intent by analysing the vocal input and matching it with an action or behaviour [29].
While ASR models have their foundation in speech transcripts, intent detection models are trained on large labelled datasets that place a label representing an action to be performed on the spoken and transcribed language.

3.4 Training a Speech Model

When training a speech recognition model, the first step is to start with the data and its preparation. As already mentioned, we remember that among the data at our disposal, there is the path, i.e., the file path pointing to the audio file and its transcript. Both these data, before being fed to the neural

network, need to be pre-processed so that they can be interpreted by the neural network.

With regard to the audio signal, which is basically a time wave, it is divided into small segments, and its frequency content is analysed. Through this process, it is then possible to generate matrices of numbers representing the time signal and its frequency. Among the possible representations that can be extracted from this conversion, there is the spectrogram, which represents the intensity of the audio signal spread over different time frequencies. Another widely used option is the *Mel Frequency Cepstral Coefficients (MFCCs)*. Such coefficients capture the acoustic characteristics of speech by transforming the audio signal into an easily interpretable sequence of data for the model. An example can be seen in fig.: 3.5.

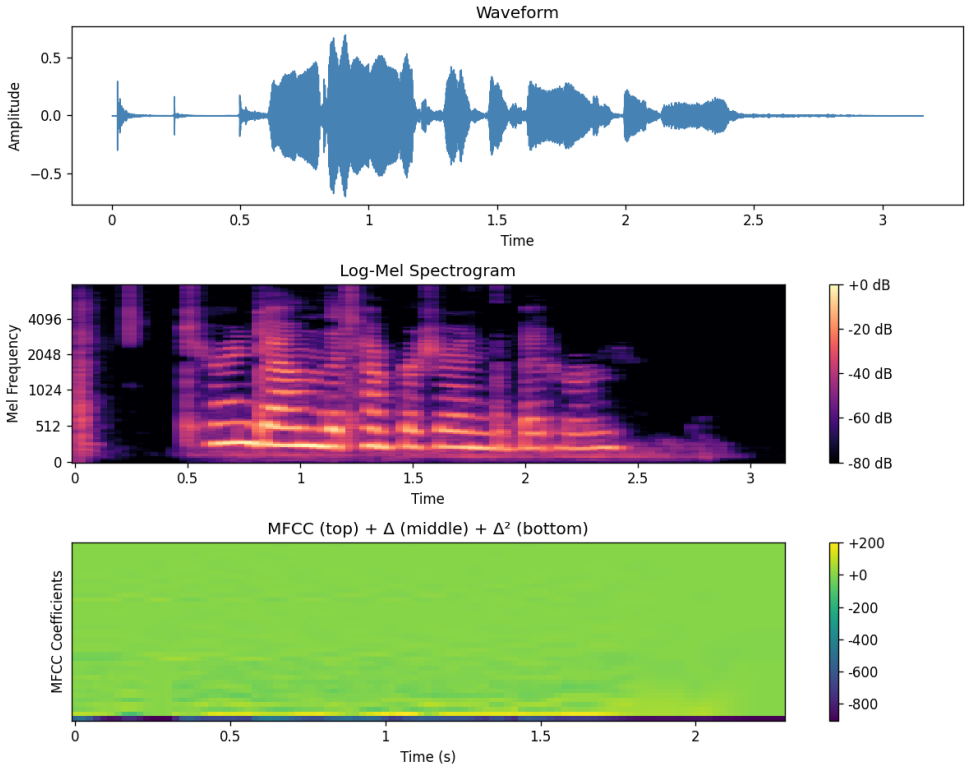

Figure 3.5: MFCCs graphs for the sentence: *"turn down the temperature in the bedroom"* of the speaker *2ojo7YRL7Gck83Z3* of the FSC dataset.

Since the model cannot understand and associate characters with phonetic segments, the transcription of the spoken text must also be transformed.

Therefore, a vocabulary is created, mapping each character or token to a unique integer. In effect, this vocabulary covers all the letters of the alphabet, numerals, and special symbols, if any, along with the `UNK` unknown token that is used to handle characters or tokens not included in the vocabulary.

For example, taking the following string: *'The smart watch costs $430,'* and a vocabulary containing all the letters of the English alphabet (lowercase), the space, the comma, and the `UNK` token, we would obtain the following representation after transformation: *'the smart watch costs i want it.'*

There is one further problem that needs to be dealt with, which is the handling of sequences with variable lengths. By this, we mean those audio inputs and their transcripts that can have very different lengths, depending on the duration of the input and the complexity of the speech. This can be handled by two techniques: padding and truncation.

With padding, the shortest sequences are filled with a generic neutral value, usually zeros, until a consistent length is achieved. Padding is very useful if data is to be processed in batches, i.e., in groups, as it allows the model to group values with the same length.
In contrast, truncation is the opposite of padding since this method cuts excessively long sequences to a predefined maximum length and decreases the computational requirements for training the model.
Furthermore, a speech recognition model can be trained either from scratch or from a template previously developed. Creating a model from scratch requires randomly initialised parameters and weights and necessitates a very large and diversified dataset in addition to a considerable amount of computing power.
If instead, one chooses to start from a pre-trained model, this is referred to as fine-tuning. Through fine-tuning, we can train the model on datasets other than the one on which it was initially trained, perhaps even further optimizing it for tasks that were not intended for the initial model.

3.5 Evaluation Metrics

It should come as no surprise to know that it is important to measure the accuracy of any machine learning system. Whether it's a self-driving car, a Natural Language Understanding (NLU) system like Amazon Alexa, or an Automatic Speech Recognition system, if one does not know how accurate the machine learning system is, it's impossible to use it in a real-world

application.

Accuracy

Within the model evaluation metrics we are going to look at in this study, we are going to use accuracy. This metric is a simple indication of how often a model delivers correct predictions. In practice, accuracy is obtained by dividing the number of correct forecasts by the total number of forecasts, i.e. if the model has correctly categorised all predictions[30]. When the model has got all predictions right, the accuracy is 100%. If it has got none right, the accuracy is zero. Nevertheless, accuracy has some limitations when it is used as a metric: as we can already guess, as it takes into account all equally important classes, it assesses only the overall number of forecasts correctly without distinguishing the severity of the errors or their distribution over the forecasts as a whole. Recognising the importance of this matter, particularly with respect to the issue of recognition of intent, we will, in fact, in the course of the analysis, combine the accuracy metric with an in-depth analysis using DivExplorer. Continuing with our discussion, we need to remember that in some scenarios a high accuracy can be misleading as it may reflect a good performance of the model on a class, probably the majority, but it hides the discrete or low performance of the model when it comes to a class of data that perhaps appears more rarely. For instance, we can take a speech recognition model that may have a very high accuracy, but on unusual words and unrepresented phonies and accents it almost always fails.

Word Error Rate (WER)

Word Error Rate is a measure of how accurately an Automatic Speech Recognition (ASR) system performs, which has become the de facto standard for measuring how accurate a speech recognition model is [31]. As the name implies, it calculates how many errors are present in the transcription text produced by an ASR system when compared to the correct human transcription.

To calculate the math beyond WER, it's quite easy. One needs to combine the number of Substitutions (S), Deletions (D), and Insertions (N), divided by the Number of Words (N).

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

Considering the following sentence: *"The quick brown fox"*, if our ASR system is not very good, it will predict the following transcription: *"The quick brown box"*. In this case, the WER would be 25%. That's because there was 1 Substitution: "fox" was substituted with "box." In this case, the ASR system predicted only: *"The quick"*, and for some reason didn't even predict the remaining words, our WER would be 50%. This is because there are 2 Deletions—only 2 words were predicted by our ASR system when 4 words were actually spoken.

The lower the Word Error Rate, the better the performance of the ASR model. One can think of word accuracy as $1 - \text{WER}$. So, if the Word Error Rate is 20%, then the word accuracy, i.e., how accurate the transcription is, is 80%.

Chapter 4

Methodology

In the following chapter, we will provide a high-level overview of how we will investigate the incremental performance of groups during the training of speech recognition models. This will be accomplished over three distinct phases: the fine-tuning of the model, the extraction of results and metadata and finally the analysis of the subgroups.

Fine-tuning and Assessment

The first phase involves the fine-tuning of a previously trained speech recognition model on a user-specified dataset. This can occur for both intent recognition and ASR tasks. During its training, the model saves both itself and its progression in a predetermined number of checkpoints.
Those checkpoints, whose performance and accuracy gradually increase over time, will then serve as the basis for model evaluation and subgroup extraction. In fact, it is through these checkpoints that we can gain an insight into the evolution of the performance of the model itself through their analysis, and have a concrete possibility of seeing where it undergoes noteworthy variations.

In the second phase, we will use the previously saved checkpoints to use them incrementally as predictors for the same dataset on which they have been fine-tuned. This process, once used to predict both the test and training dataset, sees its results already containing the metadata of the dataset enriched with additional metadata that is extracted from the context of the recordings.

Analysis and Subgroup Monitoring

The third phase eventually involves the subgroup analysis that will enable this study to incrementally monitor the evolution of sub-group performance during the training of a model on a given dataset. Initially, this phase will take the results obtained from phase two and for each pair of results obtained, i.e. training and testing, will discretise the quantitative variables if present. For instance, if the results contains the variable 'duration of speech', which we might assume ranges from 3 to 12 seconds, the program will discretize them into three distinct bins such as: 3-6, 7-9, 10-12.
When the discretization is complete, we use DivExplorer to identify the subgroups via n attributes specified by the operator. Such attributes must be present within the result dataset and may be, for example: mother tongue, gender, age or length of speech. The subgroups that have a minimum support of 5% are identified using DivExplorer.
Having found the subgroups, they are then grouped by Jaccard similarity. If the subsets are at least 80% similar, they will be merged to reduce the complexity of the subsequent analysis.
Through this procedure, which as we have described will elaborate both training and testing subgroups for a given checkpoint, it will then be possible to group them in order to obtain an overview of how they develop over time.
Indeed, once the results of all the checkpoints have been processed, we select the five best and worst subgroups by performance for the first checkpoint and then use a series of graphs to study their development over time.

Tools and Frameworks

The implementation of the models relied on popular machine learning frameworks:

- **PyTorch** was used for model training, offering flexibility in defining and optimizing neural network architectures.

- **NumPy** and **Pandas** were employed for data preprocessing and manipulation, enabling efficient handling of large datasets and metadata.

- **Matplotlib** was used for visualizing training metrics, including loss and accuracy curves, to monitor the model's performance over time.

4.1 Dataset Description

In the following, we will examine which datasets were selected to be used in this study. Such datasets are considered standard amongst the literature and provide us with the opportunity to cover a variety of linguistic and demographic characteristics. The selection of these datasets, in fact, will indeed be instrumental in helping us during the analysis of the disparities between subgroups and to assess their evolution during training.

Fluent Speech Commands (FSC) Dataset

The Fluent Speech Commands (FSC) dataset [32] comprises real audio recordings designed for spoken language understanding tasks, specifically geared toward controlling smart-home appliances or virtual assistants. The dataset includes 30,043 utterances recorded by 97 speakers, all in 16 kHz single-channel `.wav` format, with each file containing a single command such as *"put on the music"* or *"turn up the heat in the kitchen"*. These utterances are labeled across three semantic slots: *action*, *object*, and *location*. Each slot can take on various values; for instance, the *location* slot might be *"none"*, *"kitchen"*, *"bedroom"*, or *"washroom"*. The combination of slot values constitutes the intent of the utterance. For every intent, multiple verbal expressions can be mapped to it. For example, the intent {*action: "activate", object: "lights", location: "none"*} may correspond to phrases such as *"turn on the lights"*, *"switch the lights on"*, or simply *"lights on"*. In total, the dataset includes 248 distinct phrases mapping to 31 unique intents.

SLURP Dataset

The SLURP (Spoken Language Understanding Resource Package) dataset [33] is a collection of spoken English audio files spanning 18 domains, which is substantially larger and linguistically more diverse than previously existing datasets. It is designed for the development of spoken language understanding systems, particularly for end-to-end (E2E) models. It contains 72,000 audio recordings of single-turn user interactions with a virtual assistant in various in-home settings, captured in typical home or office acoustic environments. The dataset is annotated at three levels of semantics: *Scenario*, *Action*, and *Entities*, covering 18 scenarios, 46 defined actions, and 55 unique entity types.

An example of a labelled utterance might include a command such as *"Make a calendar entry for brunch on Saturday morning with Aaronson"*, which

is annotated as: {*Scenario: "Calendar", Action: "Create entry", Entities: [event name: "brunch", date: "Saturday", timeofday: "morning", person: "Aaronson"]*}.

LibriSpeech Dataset

The LibriSpeech dataset [34] is a large-scale collection of English speech derived from public domain audiobooks provided by the LibriVox project. It contains approximately 1,000 hours of spoken English at 16 kHz audio and is specifically designed for automatic speech recognition systems. LibriSpeech audio data is aligned with the corresponding text from the audiobooks and is partitioned into subsets based on the quality of the audio, such as *"clean"* and *"other"* categories. The *"clean"* subset consists of audio with relatively high recording quality and speakers whose accents are closer to standard American English, while the *"other"* subset includes more challenging audio, often with background noise or less standard accents.
The dataset is already divided into training, test, and validation sets, ensuring no speaker overlap between sets, with approximately 40 hours of development and test data combined and over 900 hours of training data across various subsets.

ITALIC: An Italian Intent Classification Dataset

The ITALIC dataset represents the first audio data collection specifically designed for intent classification in spoken Italian. It was introduced by Koudounas et al. [35] and includes both text transcripts and voice recordings, each annotated with a total of 60 intent categories for the development of spoken language understanding systems in Italian.
This dataset is based on the MASSIVE NLU collection [36], a popular dataset that provides a set of annotated textual sentences for 60 types of intent. From this basis, over 70 native Italian-speaking volunteers were then recruited to record the dataset sentences in their voices. Beyond the creation process of this dataset, the speakers also provided additional information about themselves, such as:

- **age**: the age of the speaker.
- **is_native**: whether the speaker is a native Italian speaker or not.
- **gender**: the gender of the speaker, self-annotated.

- **region**: the region of the speaker, self-annotated.
- **nationality**: the nationality of the speaker, self-annotated.
- **lisp**: any kind of lisp of the speaker, self-annotated. It is empty in case of no lisp. *(a lisp is a speech defect in which s is pronounced like th in thick and z is pronounced like th in this)*
- **education**: the education level of the speaker, self-annotated.
- **environment**: the environment of the recording, self-annotated.
- **device**: the device used for the recording, self-annotated.

Dataset	# Utterances	Hours	# Speakers	Language	Domain	Notes
FSC	30,043	19 h	97	English	Smart-home commands	Age/gender
SLURP	∼72,000	58 h	177	English	Virtual assistant	Close/far mic
ITALIC	16,521	15.46 h	70	Italian	Voice assistant	Region, age, device
LibriSpeech	∼281k segments	1,000 h	∼2,456	English	Audiobooks	Public domain

Table 4.1: Datasets overview

4.2 Models Overview

This section offers an overview of the main speech recognition models used in this research, in particular, the models developed by Facebook, such as wav2vec 2.0 and HuBERT.

wav2vec 2.0 base

wav2vec 2.0 base is a self-supervised speech learning model capable of learning textual representations from audio sampled at 16 kHz [37, 38]. As the model was trained via self-supervised learning, it does not include a tokenizer, as its training was carried out exclusively on audio data without the inclusion of textual data. Hence, it is necessary to create a custom tokenizer and then subsequently fine-tune the model using textual data to be transcribed and then labeled by the tokenizer.
Additionally, the wav2vec 2.0 model processes the raw signal through a convolutional encoder that extracts its characteristics and then passes it to a transformer that captures its dependencies. Although this process may seem complicated at first glance, it is the separation that takes place in the learning process from the acoustic representation to the decoding part of the text. Because of this, the model can perform reasonably well even when the amount

of textual data at its disposal is very limited; for example, if we only have about ten minutes of labelled data, the model will still be able to achieve very low errors following fine-tuning.
During the experiments made by the researches at Facebook it was see that this model could achieve with LibriSpeech, up to 1.8-3.3% on WER.

Wav2vec2-xls-r-300m

Wav2vec2-xls-r-300m is an extension of the aforementioned wav2vec 2.0, as this model specializes in a multilingual speech recognition domain [39, 40]. Like its predecessor, this model was trained through self-supervised learning with 436,000 hours of unlabeled audio from various sources such as VoxPopuli[41], MLS[42], CommonVoice[43], BABEL[44], and VoxLingua107[45] and comes to cover as many as 128 languages.
As this model is only trained on speech representations, it does not integrate a tokenizer, so to use it for tasks such as speech recognition, a fine-tuning step is required first to associate it with labelled data.
As for performance, the model reduces WER errors by up to 20/33% on average on datasets such as Babel, MLS, and VoxPopuli when compared to similar models; furthermore, its cross-language performance is good enough to be useful in tasks such as translation from English to other languages.

wav2vec2-large-xlsr-53-italian

The wav2vec2-large-xlsr-53-italian model is a variant of wav2vec 2.0-large that is fine-tuned to recognize the Italian language [46]. As the previous model, this one is based on self-supervised learning and requires a tokenizer when used. In detail, this model was refined using the Common Voice 6.1 [47] dataset for the Italian language and, as the industry standard dictates, was designed to process audio signals sampled at 16 KHz.

hubert-base-ls960

Hubert-base-ls960 is a model created through self-supervised learning for speech learning based on the Hidden-Unit BERT (HuBERT) method [48, 49]. This methodology uses a clustering step (such as k-means clustering) to generate labels via a semi-supervised process that allows the model to still learn audio representations without the labelled data. This approach allows the model to simultaneously learn both the acoustic characteristics of certain aspects that are structural to language and allows it not to rely on a predefined lexicon, thus differentiating itself from other models.
Similar to the previous models, since the model has been trained exclusively

on audio data, it does not include a tokenizer, so it will be necessary to fine-tune and associate a tokenizer with the model to use it.

hubert-large-ll60k

Model hubert-large-ll60k represents the large version of model hubert-base-ls960 [50]. It was trained on the LibriLight [51] dataset and, like its counterpart, uses a clustering mechanism to generate supervised labels that are then used in prediction. With its greater depth and a much larger number of parameters, the large version is not only more accurate but can capture deeper relationships and dependencies in the audio signal. The model does not have a tokenizer, and the results obtained with this model demonstrated a significant reduction in WER errors on benchmarks such as LibriSpeech[34] and LibriLight[52].

4.3 Implementation

4.3.1 Intent detection - Fine-tuning

The model training pipeline begins by taking the parameters passed in by the user before setting the GPU as the computing device via CUDA if it is available, else the CPU is set. The execution logic is followed by a series of constraints which will load, from the arguments, the user's specified model as well as the specified dataset. As far as any intent detection model trained with the FSC and SLURP datasets the pipeline works as follows:

First, the requested database is loaded into memory by customized methods. This methods are interchangeable and are used throughout the entire program. They return the following variables: a trio of datasets (of which the Train, testing and validation dataset), the number of labels and both label to ID and ID to label mappings.
After this step, the pipeline loads the user specified model by passing the related string parameter into "AutoModelForAudioClassification" from the transformers library. This function recognizes the string argument and automatically loads the target model with its processor.

Next, the programme subsequently estimates the class weights by first counting the frequency of occurrence of each label in the training dataset then assigning a relevancy factor to each label using the inverse of its frequency. The function returns class weights as a PyTorch tensor.
After the class weights have been calculated, the programme proceeds to dynamically define the training arguments in a function were will be defined the

training parameters that the model will then use during its training. This function has many parameters as it aims to allow the user to specify many of the training parameters directly from the command line.

Within the plethora of parameters that can be used in this feature to customise the training and properties of the model itself, we find:

- **output_dir**: The output directory where the model predictions and checkpoints will be written.
- **eval_strategy**: The evaluation strategy to adopt during training. Possible values are:
  - `"steps"`: Evaluation is performed (and logged) every `eval_steps`.
  - `"epoch"`: Evaluation is performed at the end of each epoch.
- **per_device_train/validation_batch_size**: The batch size per GPU/CPU for training and validation.
- **gradient_accumulation_steps**: The number of update steps to accumulate the gradients for, before performing a backward/update pass.
- **learning_rate**: The initial learning rate for the AdamW optimizer.
- **max_steps**: The total number of training steps to perform. For a finite dataset, training is reiterated through the dataset (if all data is exhausted) until `max_steps` is reached.
- **warmup_steps**: The number of steps used for a linear warmup from 0 to `learning_rate`.
- **save_steps**: The number of update steps before the model performs a checkpoint save.

After defining the training arguments and inserting them into the model, the programme proceeds with the training of the model. When finished, the programme will evaluate the model's performance and save the final model locally.

4.3.2 Intent detection - Inference and extraction

The section related to intent detection and inference is the combination of a file already in the initial repository of the thesis, the purpose of which is to extract metadata from a dataset, and a pipeline developed specifically for the inference task. In this pipeline, the inference tasks of saved models are executed incrementally during training, while simultaneous extraction of data occurs, preparing it for subsequent analysis.

Similarly to a standard fine-tuning approach, this module contains a number of user definable arguments that enable the programme to behave dynamically to the various challenges of the project.

The logic of this process starts similarly to the previous one: it first processes the user input parameters and then determines whether to use a GPU or CPU based on availability. Next, the programme enters a pipeline specific to the user's requested database and then begins the inferring and extracting phase. Initially, as in the previous file, the read_data function is called in order to load the various datasets into memory, then the programme proceeds to analyse the checkpoints. This is done on either all the saved checkpoints or on a specific one depending on the flag set by the user among the program parameters.

After entering the function which analyses a checkpoint individually, the program proceeds to load the saved checkpoint into memory via the path provided by the user.
As a second step, specified training arguments are defined to avoid training the model and use it only to extract metrics and to test it on the test data. A trainer is then initialised via the previously loaded model, which is then used to predict the entire test dataset.
These predictions that are currently in numerical format are converted in a few lines of code from numbers to string literals by taking the intent that the model found to be the most likely prediction.
It then calculates the accuracy percentage between the intents that the model correctly predicted and those that it got wrong. Lastly, the programme saves the predicted intent, the actual intent, the correctness and any metadata in a .csv file.

4.3.3 Automatic Speech Recognition - Fine-tuning

As far as fine tuning for ASR is concerned, the program makes use of a secondary pipeline specifically adapted for the purpose, which will now be

outlined. First of all, a dataset such as LibriSpeech is loaded into memory by calling the relevant function. This dataset is then converted into a pandas dataset and then a function is called on it to remove its special characters. The removal of special characters such as the question mark, the full stop, the exclamation mark and the semicolon is crucial in an ASR environment. This is because, as previously mentioned, each character in the text needs to be encoded into a number so that the model will then be able to recognise it. Special characters that we retain the apostrophe, this is because the apostrophe produces a very special sound in English that changes the phonetics of spoken words. On the other hand, we do not consider commas, which do not change the phonetics and could only confuse the model in its transcription task.
The pipeline then proceeds to create a tokenizer for the model. The tokenizer includes all the vocabulary, space, apostrophe and four other special characters which are then numbered to create a correspondence between the model's predictions and the actual characters. Among the special characters we have: <pad>, which is the padding token used to align sequences of characters of different lengths, <s> and </s>, which are respectively the start and end tokens of a sentence, and finally <unk>, a token representing a unknown character in the event that, for example, we accidentally forget to remove the commas from the text and the model does not know what to assume for the comma.
After creating the vocabulary and processor, the next step is to filter out all elements of the training and validation dataset that are deemed too long. This is done via a parameter passed in by the user which is set to 15 seconds by default. The compute_metrics function is then defined, a function that will then be passed to the model to calculate the Word Error Rate (WER) for each transcript predicted by the model.
Subsequently, the model designated for fine-tuning is loaded into memory based on the user-specified parameters.
At this stage, the function freeze_feature_encoder is called. This function will disable the gradient computation for the feature encoder so that its parameter will not be updated during training.
Once this is done to avoid training a finished model, the programme goes on to define the training arguments and define the trainer, which in turn after saving both the processor and the tokenizer will proceed to train the model itself.

4.3.4 Automatic Speech Recognition - Inference and extraction

As far as the inference step and data extraction for ASR is concerned, it can be said to be quite similar to the intent detection pipeline. Here too, for each saved model checkpoint, the script goes to load the model into memory and then uses it to predict both the test and the Train dataset. Once the inference is finished, the script then goes on to combine each data frame with the metadata of the source dataset. When done, the programme will save the .csv files in memory ready to be analysed.

4.4 Analysis

As far as the analysis is concerned, we make use of a thirt script to facilitate the analysis of the data produced by the inference and metadata extraction step. In detail, at first, the code processes the different .csv files representing the predictions of the incremental model during its training phase on a specified dataset. These files include a variety of information such as metadata, true label and predicted label, the accuracy as well as the Word Error rate, which is only present in an ASR context. There can be two kinds of files of .csv files for the program to analyse: half will contain the model inference and metadata extraction on the test dataset, the others will be related to train dataset.
From each file, the programme calculates the accuracy and subsequently discretises the numeric columns. Such columns, which may contain for instance the duration of the audio or the average speech rate, are split into intervals in such a way that they can be used for later subgroup analysis. The script also gives the user the possibility to select the columns (the attributes) to be analysed and to find subgroups by disregarding the others, this can be done by the defining *numeric_ columns* and *attributes* variables at the beginning of the program.
Following the processing step, the code moves on to use DivExplorer by Eliana Pastor, to identify subgroups with high divergence. Using this library, the programme first defines the significant attributes to form the subgroups, such as mother tongue, gender, age as well as the columns that have been previously discretized. We then identify the subgroups with a minimum support of 5% so that significant subgroups are selected and then calculate the divergence between the accuracy of each subgroup and the overall accuracy of the model on the dataset.

Due to the very nature of subgroups, many are similar to each other; indeed, we may encounter some cases where subgroups differ in only a small detail. For instance, we might have a subgroup ABCDEF and one ABCDE. Since the metadata contained in .csv files is so large, this creates $O(2^n)$ subgroups; therefore, to perform a better analysis, we apply a clustering technique using the Jaccard similarity[53]. With this technique, we measure the similarity between subgroups, and if they have a similarity greater than 80%, they are merged into a single group.

We should also mention at this point that the pipeline that processes the subgroups is also provided with a failsafe so that if it is noticed that the sum of the support of the subgroups does not equal 1, i.e., the 100%, then the program fails. This was done to ensure the consistency of the results.

The program continues with the analysis of the subgroups through the training to extract further insights into how the model improves or worsens for specific subgroups. This is done by loading the various .csv files into memory, storing the subgroups found in the first step, and following their evolution in the remaining steps. When this is done, we select the five best subgroups, i.e. those with the least divergence in accuracy and the five worst ones those with the greatest divergence in accuracy.
Eventually, after this process, we can generate a graph showing the evolution of error divergence for the subgroups during the model training. On this graph, the solid lines represent the 5 best subgroups, while the dashed lines represent the subgroups with the worst performance.

Chapter 5

Results

As already mentioned in the previous section, the program is able to create up to four different charts per model and dataset combination. These graphs, including a linear graph and a heatmap, can show the divergence trends of subgroups during the model's trading process.
These visualizations will, in fact, allow us to identify trends of improvement or deterioration of the model on specific subgroups during its training. In fact, we can see in detail how, for example, the gender of the subgroup impacts its performance during the model's training.
Therefore, in order to avoid going into a mechanical and sterile analysis of each graph for each result, we will limit ourselves to setting out the most interesting results and peculiarities for combinations of models and datasets. However, the complete set of graphs generated by the program remains publicly available via the link given in the appendix.

5.1 Performance Metrics

To evaluate the performance of the model, the program uses the same metrics as the model being trained. If it is intent detection, then it will use accuracy, while if it is ASR, it will use word error rate. Specifically, if the input .csv file contains the True label and predicted label columns, then the code will automatically determine the classification errors and then retrieve the global accuracy and divergence for the groups. For WER, the divergence of subgroups is calculated as a function of the average WER of the subgroup in relation to the overall WER.
The program then provides the support count, which indicates the proportion of samples that fall into a specific subgroup with respect to the entire

dataset. This statement helps us to understand how relevant a subgroup may be in the overall dataset. By default, the program excludes subgroups with less than 5% support.
Finally, the divergence value is calculated as the difference between the performance of the subgroup and the overall metric of the entire dataset for a single epoch.

5.2 Subgroup-Specific Analysis

The graphs produced by the analysis are then accompanied by a series of metrics that the analysis program extracts from the .csv files together with subgroups. They include divergence and support.
Divergence is calculated as the difference between the performance of the subgroup and the overall metric of the entire dataset for a given epoch. For example, if a subgroup 'x' has an accuracy of 0.68 and at epoch number 5, the model has an accuracy of 0.75, then its divergence will be -0.07. A somewhat similar calculation occurs for war where only the fact that the WER of the subgroup is calculated as the average of the WERs of each instance of the subgroup itself.
The program also provides the support count indicating how many samples fall into a specific subgroup in comparison to the entire dataset. This value helps us to understand the importance of the subgroup itself and how relevant it may be as a whole. For example, we might find that a subgroup of women has 61% support and performs 0.02 better than a subgroup of men with 39% support and performs 0.02 worse. The program is preset to exclude subgroups with less than 5% support.
The program is also able to automatically change its internal processes based on the type of file you pass to it as input, if the .csv files have the True label and predicted label column, then it will be intent detection, and the program will work with accuracy whereas if the program notices the WER column then it will be ASR and the program will act accordingly.

5.2.1 wav2vec2-xls-r-300m and ITALIC

General disparities

First, we would like to observe the subgroups of table 5.1 evolve in graph 5.1. Here we can notice the starting subgroups, especially D, C and B, have maintained their divergence in being predicted more easily by the model, while we note that almost all of the subgroups that were initially less performing with the model trained still have a slightly lower accuracy than the others.

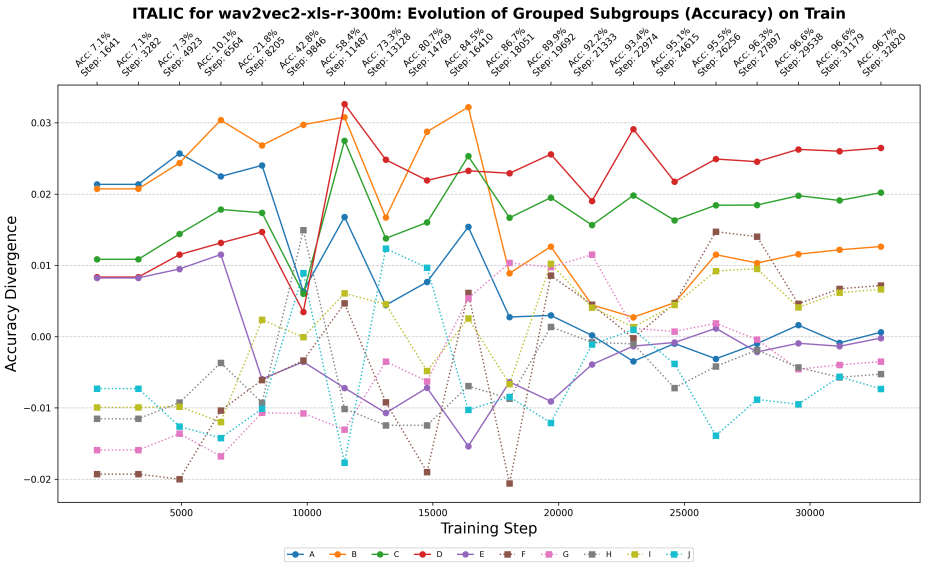

Figure 5.1: Accuracy divergence of subgroups for `wav2vec2-xls-r-300m` on the ITALIC dataset.

Letter	Group Type	Attributes	Accuracy Divergence (Delta)	Support
A	Better	education=bachelor, nationality=italiana	0.021 (Delta: -0.049)	7.40%
B	Better	device=phone, education=bachelor	0.021 (Delta: -0.050)	5.63%
C	Better	education=bachelor education=bachelor, field=close education=bachelor, lisp=nessuno is_native=False, education=bachelor	0.011 (Delta: -0.060)	11.95%
D	Better	education=bachelor, gender=male	0.008 (Delta: -0.062)	8.79%
E	Better	region=marche, environment=silent	0.008 (Delta: -0.063)	22.29%
F	Worse	region=sicilia, education=high_school	-0.019 (Delta: -0.090)	5.08%
G	Worse	education=high_school, environment=quiet	-0.016 (Delta: -0.087)	7.91%
H	Worse	education=high_school, gender=male	-0.012 (Delta: -0.082)	7.59%
I	Worse	region=sicilia, device=phone	-0.010 (Delta: -0.081)	10.48%
J	Worse	device=computer, region=piemonte	-0.007 (Delta: -0.078)	7.03%

Table 5.1: Mapping of Letters to Grouped Subgroups for Train for `wav2vec2-xls-r-300m` on the ITALIC dataset

These include subgroup A representing male university graduates, subgroup J representing speakers from Piedmont who record from their computer and subgroup E representing speakers from the Marche region who record in a silent environment. Moreover, we can observe how the performance of the subgroups J, H and G, which were slightly underperforming at the beginning of the model's training, later improved to having near no divergence. This subgroups contain different speaking accents, lower levels of education and different recording devices. Their poorer performance of these subgroups may

therefore be explained by their underrepresentation or by their having acoustic characteristics that the model found during its training more difficult to generalise.

Audio-related subgroup disparities

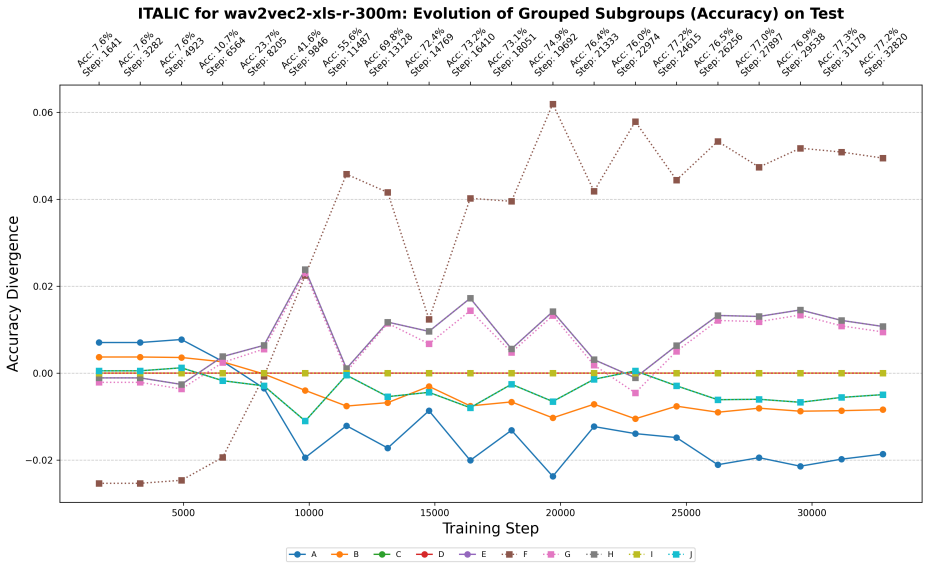

Figure 5.2: Accuracy divergence of subgroups for `wav2vec2-xls-r-300m` on the ITALIC dataset.

In the graph 5.2 of wav2vec 2.0 trained on ITALIC, one can observe that the subgroups all follow approximately the same trend. Of these, subgroups I and B are even on the zero line as a divergence so small that it can be interpreted as 0. Of these two subgroups, the B line represents (device=phone) and (field=close, device=phone) while the I line represents (field=close). We can therefore assume that as far as the test dataset is concerned, the pattern is hardly influenced at all by sounds that are recorded close up, perhaps from a telephone.
A particularly interesting element of this graph is the dashed line F, which starts with a divergence of -0.02 and ends with a divergence of approximately 0.05. The F line in this graph represents the following subgroups: (device=computer), (environment=silent), (field=close). From this information we can derive that, users recording their voice from a computer in a silent environment where the microphone is close to the speaker, a better audio quality can be produced.

Regional and gender-based disparities

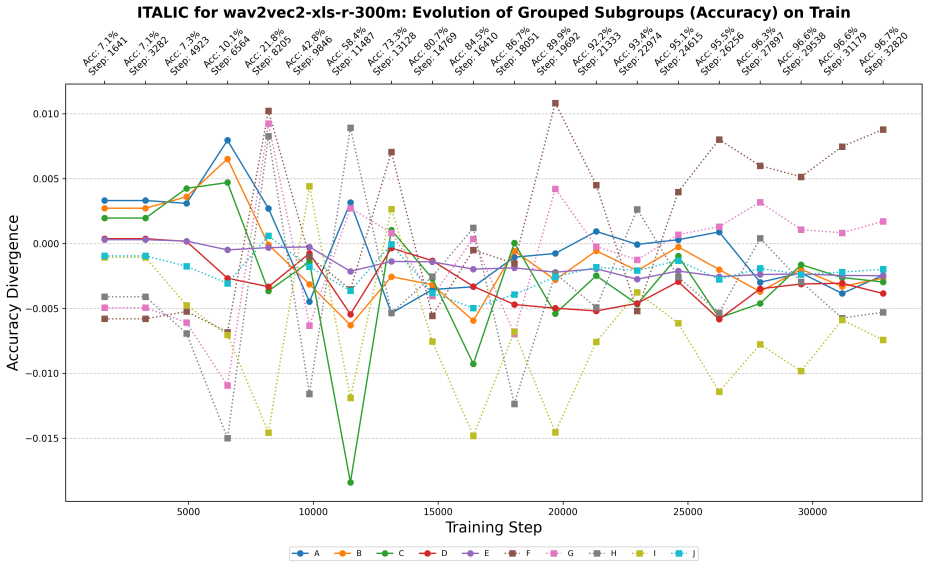

Figure 5.3: Accuracy divergence of subgroups for `wav2vec2-xls-r-300m` on the ITALIC dataset.

Changing our focus instead onto any bias the model may have on the people themselves requires a closer observation of how wav2vec 2.0 behaves with the ITALIC dataset on the Train dataset. Here in graph 5.3 we are working with very small differences in accuracy, something quite good as it conveys the information that the model is not actively discriminating against any subgroup of people.
Subsequently, we do find small differences between them, the line F for example representing the subgroup: (gender=female, region=sicily).
At first the Sicilian women were marginally penalised by the model, but they found a slight favour after the model was fully trained. Should this research make any assumptions we might in fact say that the peculiar cadence of Sicilian speech sounds better, thereby gaining some favour in the vocal recognition of x. Supporting this claim, the G subgroup, representing persons living in Sicily, initially have a slight disadvantage like the F subgroup but then with the trained model they perform slightly better than the others.

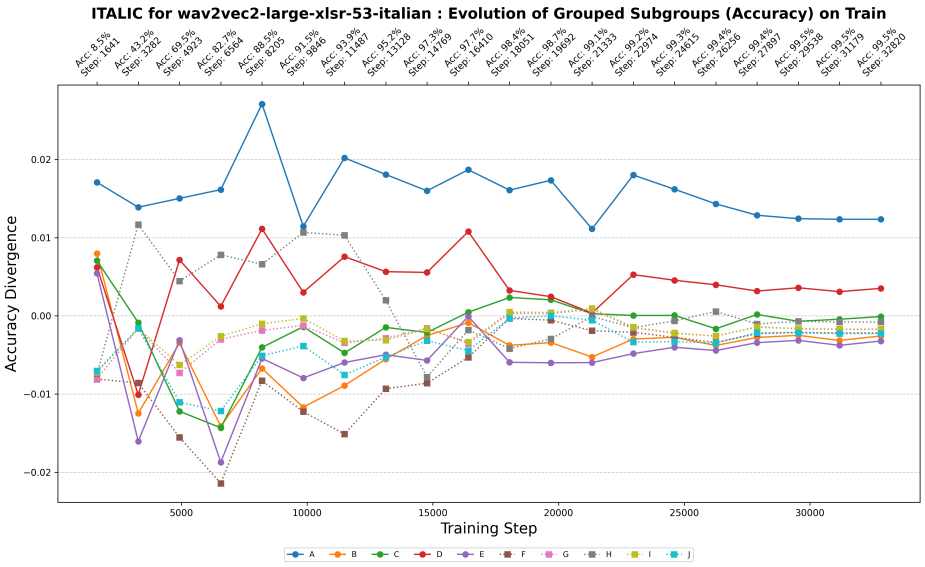

Figure 5.4: Accuracy divergence of subgroups for `wav2vec2-large-xlsr-53-italian` on the ITALIC dataset.

5.2.2 wav2vec2-large-xlsr-53-italian and ITALIC

General disparities

With regard to the wav2vec 2.0-large-xlsr-53-italian model trained on ITALIC, one can observe a general convergence of the subgroups towards a divergence of accuracy close to or equal to zero. Indeed, all three previously illustrated graphs for wav2vec2-xls-r-300m were generated and no noteworthy cases were found.
The only feature worth noting is in graph 5.4, where it can be seen that subgroup A, containing those with a university degree is best understood by the model throughout its development.

5.2.3 facebookhubert-large-ll60k and LibriSpeech

General disparities

We now illustrate the model facebookhubert-large-ll60k on LibriSpeech. Being that for dataset LibriSpeech we only have gender as metadata the analysis may have turned out to be inconclusive but nevertheless, analysing the evolution of the model and the total subgroups in the training and test dataset we can make some points that are very relevant to this research.

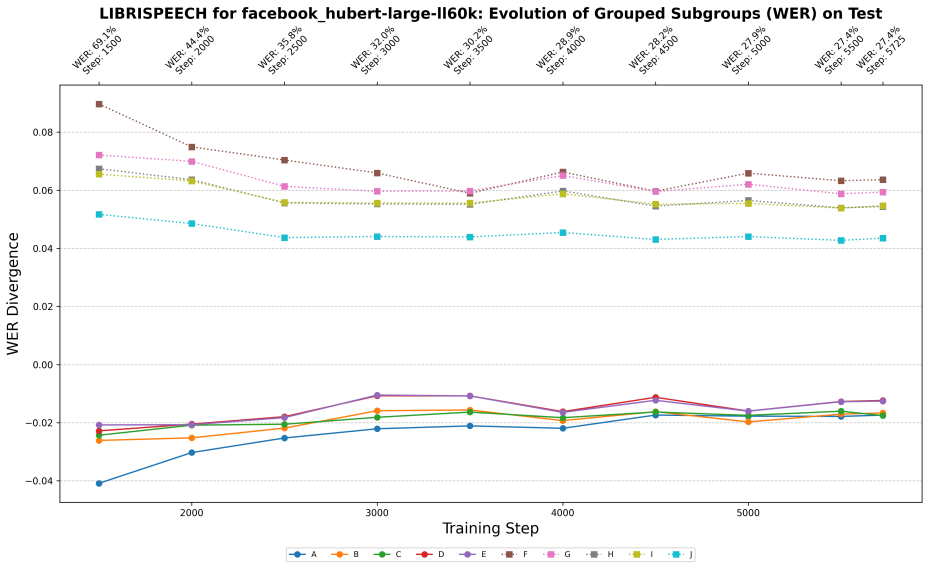

Figure 5.5: WER divergence of subgroups for `facebookhubert-large-ll60k` on the LibriSpeech dataset.

In fact, we note that when it comes to evaluating the test dataset in image: **??**, the worst subgroups never manage to improve, remaining at +0.05% WER while the subgroups with the best performance are stable at around -0.02% WER.

As to the performance of the subgroups in the Train dataset, as seen in image: 5.6, we can see that the worst subgroups have a fairly similar performance trend throughout the training of the model, something that cannot be said for the best subgroups, three of which, represented by the line E, D and A, in fact deteriorate their performance by between 0.01% and 0.03%. These subgroups have in common that they have many words: n_words_bin=31-62.

5.2.4 Facebook hubert-base-ls960 and LibriSpeech

General disparities

Considering instead the Facebook hubert-base-ls960 model and analysing all possible disparities, we note through the graphs and data at our disposal that the spread of disparities between this model and the previous one is extremely similar. This is reinforced by the fact that, as we can see from the tables 5.2

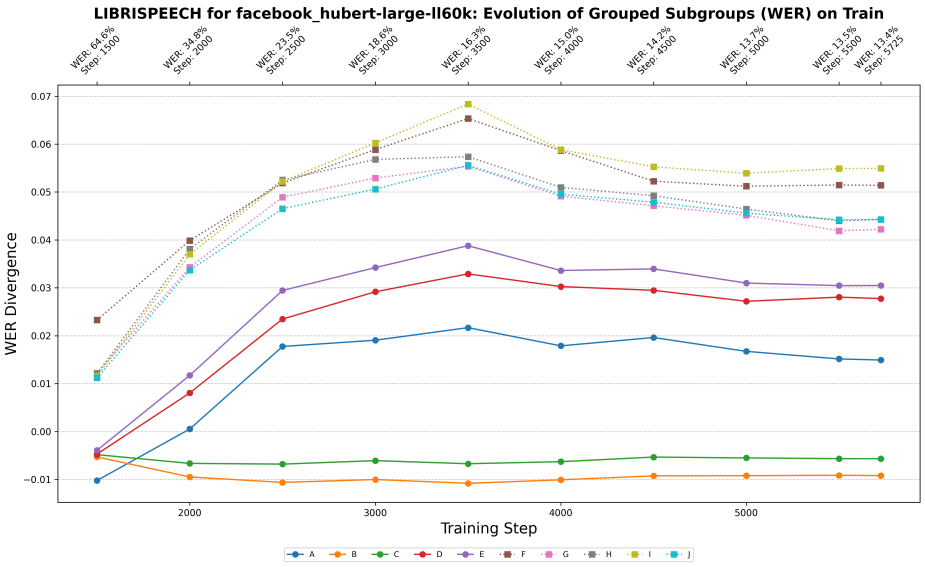

Figure 5.6: WER divergence of subgroups for `facebookhubert-large-ll60k` on the LibriSpeech dataset.

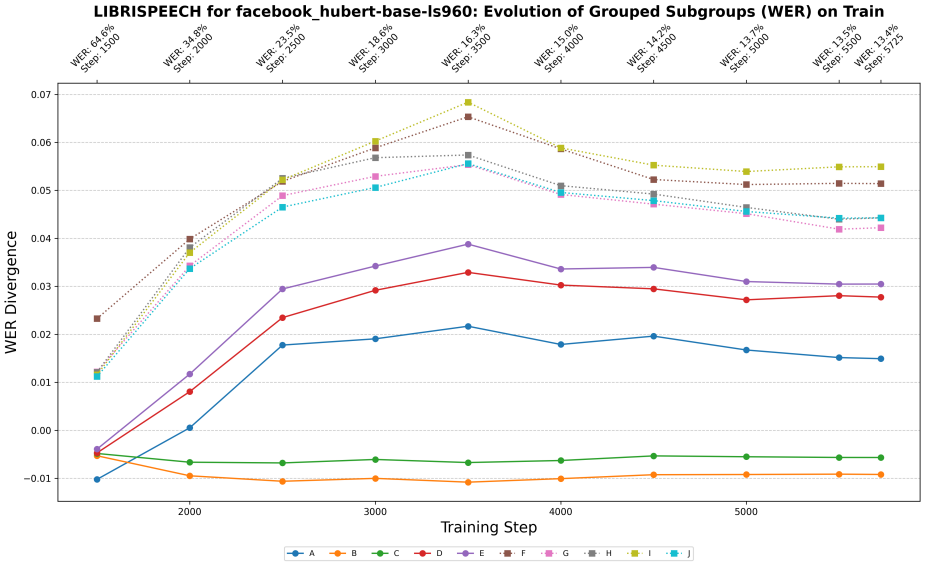

Figure 5.7: WER divergence of subgroups for `facebookhubert-base-ls960` on the LibriSpeech dataset.

and 5.3 that the subgroups and their related support is very if not exactly the same. We can therefore say with certainty that for the same model, the large and basic versions have no substantial changes in the evolution of

Letter	Group Type	Attributes	Initial WER (Change)	Support
A	Best	total_silence_bin=0-4, n_words_bin=31-62	-0.010 (Improvement: 0.656)	0.10%
B	Best	speed_rate_word_trimmed_bin=0-1, total_duration_bin=1-12 trimmed_duration_bin=0-11, speed_rate_word_trimmed_bin=0-1	-0.005 (Improvement: 0.651)	0.85%
C	Best	total_silence_bin=0-4, speed_rate_word_trimmed_bin=0-1	-0.005 (Improvement: 0.651)	0.91%
D	Best	n_words_bin=31-62, gender=female	-0.005 (Improvement: 0.651)	0.09%
E	Best	n_words_bin=31-62 speed_rate_word_trimmed_bin=0-1, n_words_bin=31-62	-0.004 (Improvement: 0.650)	0.16%
F	Worst	speed_rate_word_trimmed_bin=0-1, total_silence_bin=4-8 total_silence_bin=4-8	0.023 (Excess: -0.623)	7.77%
G	Worst	trimmed_duration_bin=11-21, gender=male	0.012 (Excess: -0.634)	6.78%
H	Worst	gender=male, total_duration_bin=12-22	0.012 (Excess: -0.634)	6.50%
I	Worst	total_duration_bin=12-22, total_silence_bin=4-8 trimmed_duration_bin=11-21, total_silence_bin=4-8	0.012 (Excess: -0.634)	5.89%
J	Worst	speed_rate_word_trimmed_bin=0-1, trimmed_duration_bin=11-21 trimmed_duration_bin=11-21	0.011 (Excess: -0.635)	12.71%

Table 5.2: Mapping of Letters to Grouped Subgroups for Train (WER) for `facebookhubert-large-ll60k` on the LibriSpeech dataset.

the subgroups during their training as seen in 5.7. In addition, if we were to examine this in more detail, we can see that subgroups with the worst performance tend to have higher support when compared to subgroups with better performance5.3. A potential explanation for this phenomenon is that larger subgroups encapsulate a higher degree of feature heterogeneity, whilst the smaller subgroups are more homogeneous and have specific patterns that the model is able to capture more easily.

Letter	Group Type	Attributes	Initial WER (Change)	Support
A	Best	n_words_bin=31-62, total_silence_bin=0-4	-0.010 (Improvement: 0.656)	0.10%
B	Best	total_duration_bin=1-12, speed_rate_word_trimmed_bin=0-1 trimmed_duration_bin=0-11, speed_rate_word_trimmed_bin=0-1	-0.005 (Improvement: 0.651)	0.85%
C	Best	speed_rate_word_trimmed_bin=0-1, total_silence_bin=0-4	-0.005 (Improvement: 0.651)	0.91%
D	Best	n_words_bin=31-62, gender=female	-0.005 (Improvement: 0.651)	0.09%
E	Best	n_words_bin=31-62 n_words_bin=31-62, speed_rate_word_trimmed_bin=0-1	-0.004 (Improvement: 0.650)	0.16%
F	Worst	total_silence_bin=4-8 total_silence_bin=4-8, speed_rate_word_trimmed_bin=0-1	0.023 (Excess: -0.623)	7.77%
G	Worst	trimmed_duration_bin=11-21, gender=male	0.012 (Excess: -0.634)	6.78%
H	Worst	total_duration_bin=12-22, gender=male	0.012 (Excess: -0.634)	6.50%
I	Worst	total_silence_bin=4-8, total_duration_bin=12-22 total_silence_bin=4-8, trimmed_duration_bin=11-21	0.012 (Excess: -0.634)	5.89%
J	Worst	trimmed_duration_bin=11-21 trimmed_duration_bin=11-21, speed_rate_word_trimmed_bin=0-1	0.011 (Excess: -0.635)	12.71%

Table 5.3: Mapping of Letters to Grouped Subgroups for Train (WER) for `facebookhubert-base-ls960` on the LibriSpeech dataset.

5.2.5 hubert-base-ls960 and FSC

General disparities

As can be seen from Figure 5.8 with regard to the model hubert-base-ls960 on the FSC dataset, we can notice a very peculiar thing, namely that from the very first it is noticeable that the majority of the best and worst 5 subgroups have no if very little divergence. The only outliers, which are then immediately integrated into the learning of the model, are the subgroups F, G and H.

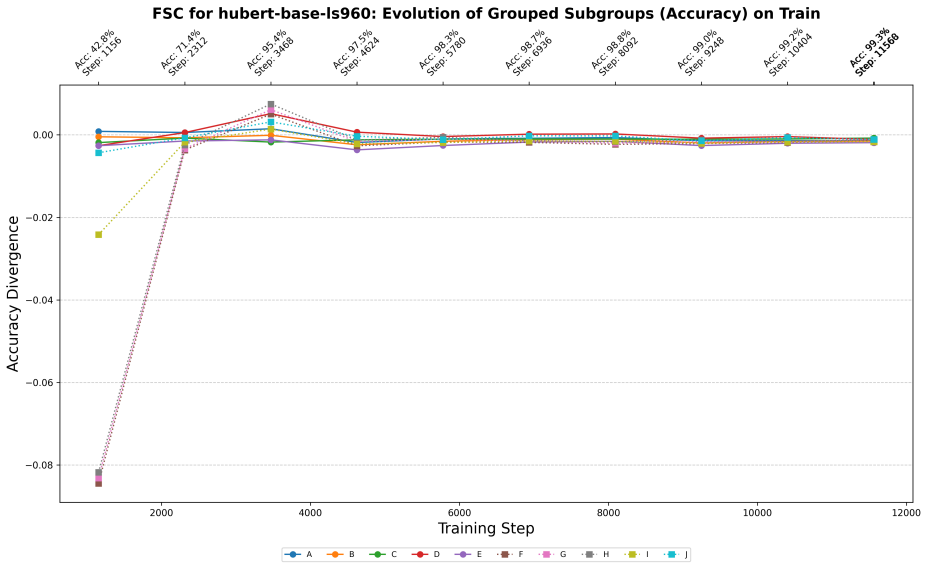

Figure 5.8: Accuracy divergence of subgroups for `hubert-base-ls960` on the FSC dataset.

5.2.6 wav2vec 2.0 and FSC

General disparities

On the other hand, as far as model wav2vec 2.0 trained on FSC is concerned, we can see in graph **??** that in the training phase the model has a different convergence on the three worst subgroups, which are still the same subgroups mentioned in the previous section.

5.2.7 wav2vec 2.0 and SLURP

General disparities

Let us now analyse the general performance of wav2vec 2.0 on the SLURP dataset, here we can see in figure **??** a very interesting general trend. When the model is accurate for only 6.6% of the cases, all subgroups start with a particularly low divergence of accuracy.
Once the model is trained, however, the subgroups that had a very slight divergence in accuracy at the first evaluation are more easily predicted throughout the training phase of the model. This is the opposite for the subgroups that had been identified as the best in the first epoch, all of which had significantly lower performance, decidedly opposite to the worst subgroups.
To better understand this strange evaluation we can then turn to the graph

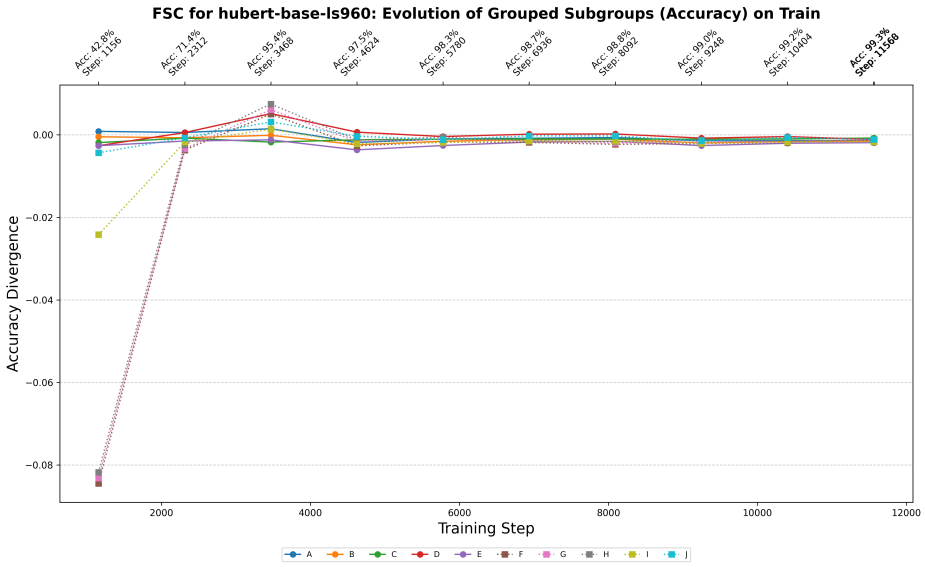

Figure 5.9: Accuracy divergence of subgroups for `wav2vec 2.0` on the FSC dataset.

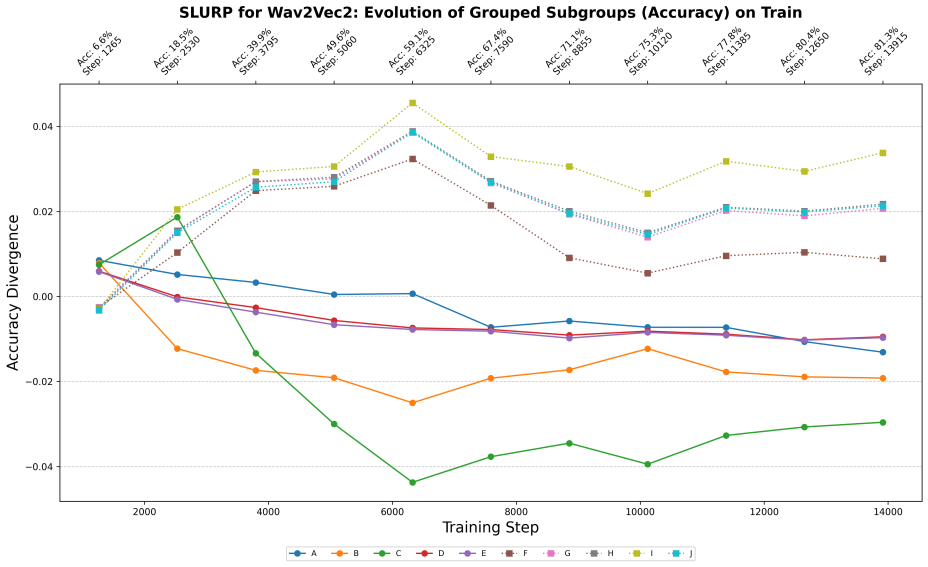

Figure 5.10: Accuracy divergence of subgroups for `wav2vec 2.0` on the SLURP dataset.

??, where we find the subgroups in the test dataset, in this case we note that the majority of the subgroups have a slightly negative performance with an average of -0.01 while we have only one outliner, the F line. The F line represents men who speak slowly.

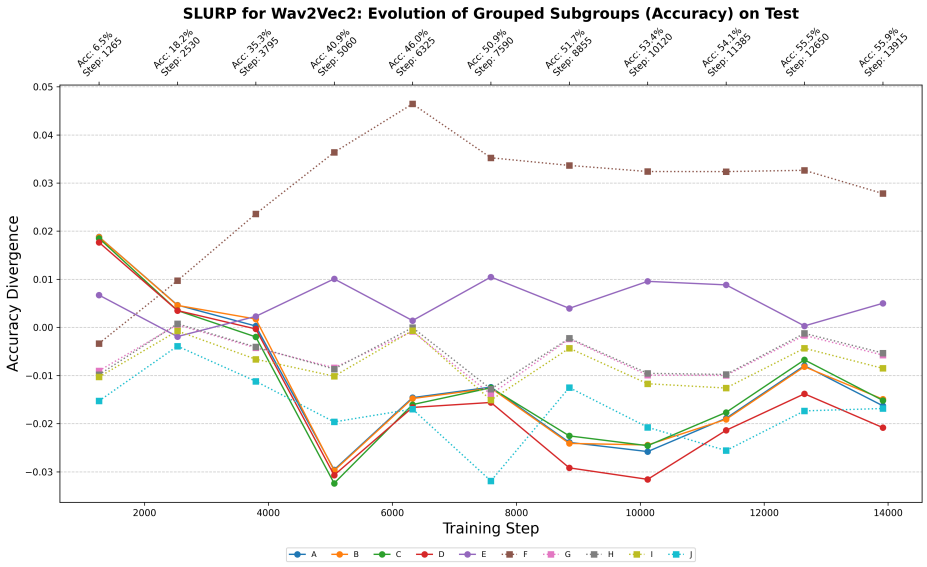

Figure 5.11: Accuracy divergence of subgroups for `wav2vec 2.0` on the SLURP dataset.

5.3 Intent classification Demographic Analysis

In this supplementary section of the investigation we will go into more detail on how the performance of the subgroups and demographics of Wav2Vec 2.0 may differ on the SLURP, ITALIC and FSC datasets. The aim of this further analysis is in fact to test whether significant differences emerge during the training of the models that may lead to disparities and discrimination on speaker demographics.

5.3.1 wav2vec2 and FSC

To begin with,as we observe Fig 5.12, we can state that where wav2vec2 is trained with FSC, female speakers have a higher accuracy index than their counterparts. In fact, subgroup F is the worst performing subgroup comprising male speakers aged between 41 and 65, which is in direct contradiction to the best performing subgroup C, this time comprising female speakers also aged between 41 and 65. When the model training is complete, virtually all subgroups except C are found to have almost no divergence. This highlights how wav2vec2 trained on FSC does not lead to the creation of subgroup demographic discrimination except for a slight bias towards older women. We can therefore assume that this particular subgroup has a clearer way of speaking and interacting with the model.

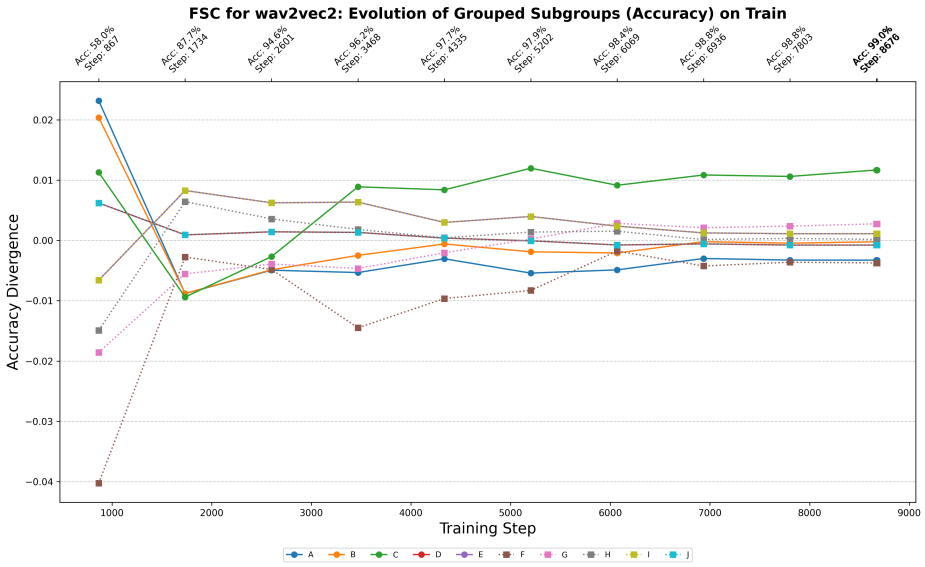

Figure 5.12: Accuracy divergence of subgroups for `wav2vec 2.0` on the FSC dataset.

5.3.2 wav2vec2 and SLURP

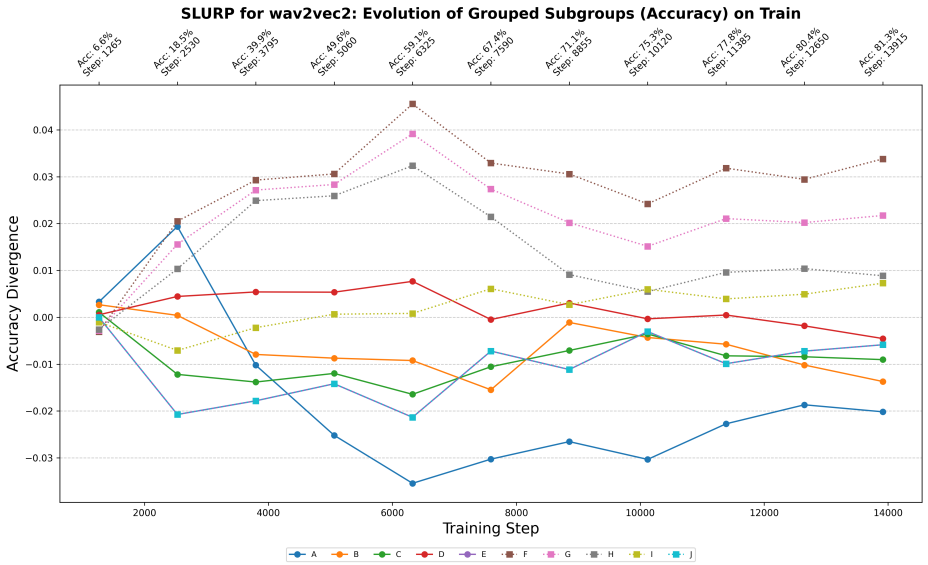

Figure 5.13: Accuracy divergence of subgroups for `wav2vec 2.0` on the SLURP dataset.

Regarding wav2vec2 trained with SLURP, we can first notice in Fig 5.13

Letter	Group Type	Attributes	Accuracy Divergence (Delta)	Support
A	Better	gender=Unknown | origin=Unknown origin=Unknown, gender=Unknown	0.003 (Delta: -0.063)	9.68%
B	Better	gender=Female, origin=Native	0.003 (Delta: -0.064)	23.25%
C	Better	gender=Female	0.001 (Delta: -0.065)	57.45%
D	Better	origin=Native	0.001 (Delta: -0.066)	39.16%
E	Better	origin=Non-native, gender=Female	-0.000 (Delta: -0.066)	34.20%
F	Worse	gender=Male, origin=Non-native	-0.003 (Delta: -0.069)	16.96%
G	Worse	gender=Male	-0.003 (Delta: -0.069)	32.87%
H	Worse	gender=Male, origin=Native	-0.003 (Delta: -0.069)	15.91%
I	Worse	origin=Non-native	-0.001 (Delta: -0.067)	51.16%
J	Worse	origin=Non-native, gender=Female	-0.000 (Delta: -0.066)	34.20%

Table 5.4: Mapping of Letters to Grouped Subgroups for `wav2vec 2.0` on the SLURP dataset.

and table 5.4 how there are big differences between the performance of male and female speakers. As a matter of fact, in this configuration the female speakers tend to be closer to or above the average while the male speakers, particularly the non-native speakers, are consistently below the average. As we do not possess information on the age of the speakers for analysis, we resort to yet another very important factor for demographics, which is whether the speaker is a native speaker or not. In this analysis, we can observe, as one might suppose, that being native speakers generally yields a slight improvement, although this is not as distinct as in other contexts. Furthermore, we can also note that the subgroup I consisting of non-native male speakers seems to show the worst combination. Lastly, we note how subgroup A, comprising speakers whose gender is unknown, starts with a slight advantage but as soon as the pattern starts to specialise it is penalised. This may also be due to the fact that subgroup A is the subgroup with the least support in the dataset so it is assumed that the model did not learn it well.

5.3.3 Wav2Vec 2.0 and ITALIC

Finally, when considering wav2vec2 and ITALIC in fig: 5.5, we immediately notice the subgroups F and G, respectively comprising males aged between 43 and 64 and speakers also aged between 43 and 64. These two sub-groups start off with a slight negative divergence in their accuracy but during the training consistently perform better than all the other sub-groups. In this case, G has 27% support while F has 9% support as seen in table: **??**.

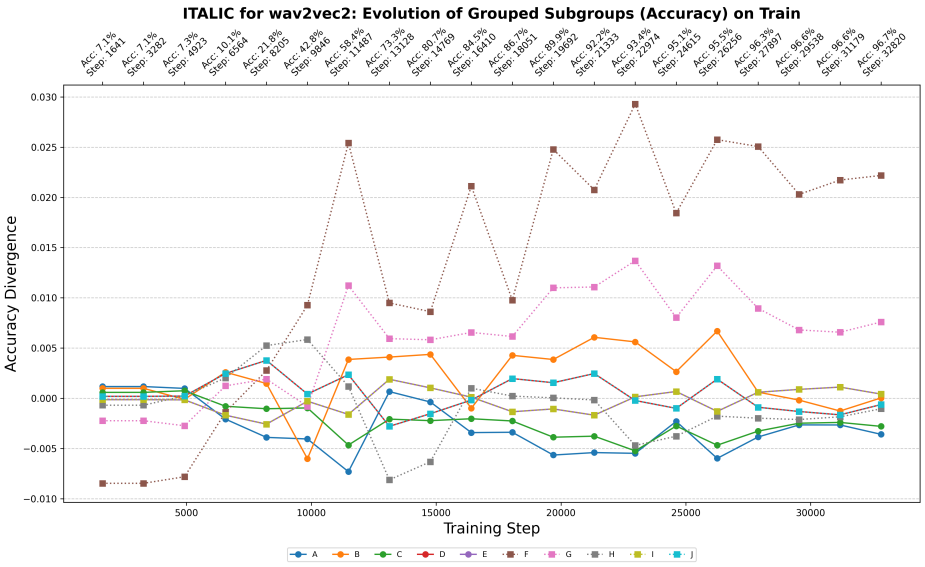

Figure 5.14: Accuracy divergence of subgroups for `wav2vec 2.0` on the ITALIC dataset.

5.3.4 Wav2Vec 2.0 and ITALIC

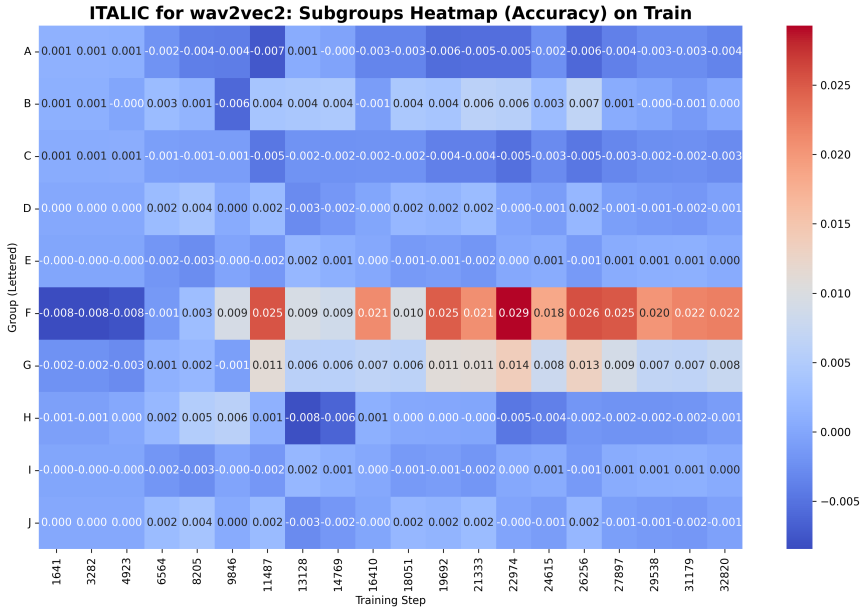

Figure 5.15: Accuracy divergence of subgroups for `wav2vec 2.0` on the ITALIC dataset.

Letter	Group Type	Attributes	Accuracy Divergence (Delta)	Support
A	Better	gender=male, age_bin=22-43	0.001 (Delta: -0.070)	49.36%
B	Better	gender=female, age_bin=43-64	0.001 (Delta: -0.070)	18.01%
C	Better	age_bin=22-43	0.001 (Delta: -0.070)	71.75%
D	Better	gender=female	0.000 (Delta: -0.071)	40.48%
E	Better	gender=male	-0.000 (Delta: -0.071)	59.52%
F	Worse	gender=male, age_bin=43-64	-0.008 (Delta: -0.079)	9.33%
G	Worse	age_bin=43-64	-0.002 (Delta: -0.073)	27.34%
H	Worse	age_bin=22-43, gender=female	-0.001 (Delta: -0.071)	22.39%
I	Worse	gender=male	-0.000 (Delta: -0.071)	59.52%
J	Worse	gender=female	0.000 (Delta: -0.071)	40.48%

Table 5.5: Mapping of Letters to Grouped Subgroups for `wav2vec 2.0` on the ITALIC dataset.

Looking in greater detail, we can see that some of the lines of the subgroups are superimposed, so we will now refer to the heatmap in figure 5.15. Here we can see that the male and female subgroups E and D have an extremely low divergence respectively. Therefore, we can conclude that the model performs exceptionally well at recognising speech without discriminating by gender.

5.3.5 Results

Observing the data in its entirety, we can observe some common patterns with respect to demographic characteristics. First among these is certainly the gender distribution, in fact, in the FSC and SLURP datasets we find that female speakers perform better than men. In ITALIC, on the other hand, we notice how gender differences tend to decrease during the training of the model. From this we can draw the conclusion that, given the balanced representation in terms of numbers between female and male speakers in the dataset, female speakers are better recognised as they are more homogenous among themselves. Furthermore, we can also conclude that older speakers, from 41 to 65 years old, have more accurate results especially if they are female. This may indeed indicate a greater clarity or slowness of speech that is very often typical for older age groups. Ultimately, we can conclude as per our assumption that non-native speakers are often penalised by a different accent or intuition in a less variant dataset.

5.4 Training Environment

The training process was conducted on Kaggle, which provided an accessible platform for model training. The experiments utilized a GPU-enabled

runtime, specifically the NVIDIA Tesla P100 GPU, which offered high computational power for training deep learning models. The Kaggle environment also facilitated the seamless integration of dependencies with pre-installed libraries and a customizable configuration that streamlined the experimental workflow. [1]
The experimental setup for this study made use of Python 3.10.14.

5.5 Analysis and Discussion

Upon observing the data produced by our analysis of the models and datasets, we can certainly conclude that the speech recognition models each show distinct patterns with regard to the handling of performance disparities between the various subgroups. This is the case for the attributes of the dataset such as gender, age and accent of the speaker as well as for other acoustic variables such as speech speed and total length of the audio file.
The trend that was most frequently detected during the incremental analysis and monitoring is that although the accuracy of the models is progressively better some subgroups follow performance dynamics that on average are slightly different from the overall average. This confirms our initial hypotheses where we assumed that model training could introduce biases and divergences towards certain subgroups.
Amongst the divergences that our analysis brought to light, it was observed that some subgroups consistently maintain systematic divergences by having a static performance gap compared to the overall dataset, while others, albeit to a lesser extent, show recovery or deterioration trends but still tend to conform towards zero divergence.
We can therefore assume with good confidence that the appearance of subgroups does not result from severe disparities in the training data but instead from structural dynamics that the models have during their training. These dynamics of performance disparities may in fact be triggered at specific points during the training of the model where even the presence of slight imbalances in the optimisation or distribution of the data may accentuate the disparities. This opens up the way for future research where direct intervention in the reallocation of model weights and targeted resampling of the data when subgroups are encountered that exceed an accuracy threshold or WER divergence.

[1] In addition, the heaviest experiments were conducted by an NVIDIA 3080TI GPU provided by Jacopo Franco Electronics PhD Student from Newcastle University, UK.

5.6 Model Complexity vs. Dataset Complexity

Continuing our analysis of the results, one can examine everything together and notice a particularly interesting fact. Indeed, we note that the more complex and deeper model architectures with a larger number of parameters (such as the 'large' models) are not necessarily going to guarantee a reduction in the subgroup performance disparity.
Indeed, we can note that although the overall performance is better in more complex models, this does not affect subgroups with minority speech patterns and accents in the same way. We may explain this particular result by the fact that large models are trained with a huge amount of data. It is precisely because of the size of these training datasets that they are often unbalanced, as we have previously seen, and the large amount of data results in a lack of diversification. From this we can therefore say that the complexity and variety of the data used to train large speech recognition models is fundamental in order to obtain balanced models and consequently fair results for all subgroups in the training dataset.

5.7 Specific Subgroup Analysis

As stated in our observation of the results, one can easily notice that the speech recognition models show very different behaviours once we examine how the disparities in the performance of the subgroups evolve during training. As a matter of fact, if we examine the same model trained with two different datasets, we can observe that the evolution of the performance of the subgroups, albeit with different subgroup scores, shows similar performance trends. This suggests that there are specific cases in which the propagation of differences between subgroups is not just a matter related to the dataset used for model training, but also with the model itself, which with its architecture and prior training adds a structural bias which then leads to the creation of disparities.
With regard to accuracy, we then notice how, although the average performance of the models has a positive trend, the performances of the subgroups display three distinct behaviours. On some occasions the performance of a subgroup begins advantaged and concludes advantaged, on other occasions a disadvantaged subgroup starts the training at a disadvantage and stays disadvantaged even when the training is finished. Then, there are a number of subgroups that experience a ''bounce-back''. One can easily trace this phenomenon back to a progressive adaptation of the model towards such speech characteristics as, for example, accent and speed which were initially

hard to predict. Furthermore, the performance of the other subgroups, i.e. those that retain or sometimes even widen their gap compared to the overall performance, show how the models have an insufficient ability to predict uncommon and non-dominant patterns while instead being more optimised on common and dominant speech in the population. As a result, unless this set of subgroups coincides both with the dominant features of the other speakers in the dataset and with the architecture and fine-tuning with which the initial model was created, it will be noted that the model tends to neglect these subgroups, having limited recovery of the less performing ones in the more advanced stages.

5.8 Limitations

With regard to the limitations addressed by this thesis, it can be seen that it is first necessary to train less performing models in order to highlight subgroup bias even more. This is because this research made use of well-known models developed by Facebook, which, being a large company, will certainly have had the opportunity to invest research and economic resources to release models that perform well on all occasions, in our case as far as subgroups are concerned. This could be related to the need to use datasets with unbalanced data not adequately representing all the real variations in the population. This choice could have a further window of study on how groups develop this time when they are unrepresentative.
Among the most important limitations that this thesis has faced is certainly a difficulty in identifying precise dynamics that then lead to systematic discrimination during model training. This problem was also exacerbated by the limited computational resources for model trading and the limited possibility of scaling up experiments to test various combinations of datasets and models.

5.9 Improvements and Recommendations for Future Work

Furthermore, concerning the improvement strategies (and recommendations for the future), the following should definitely be considered. First and foremost is the leveraging of subgroup analysis when training the model to comprehend and correct in real-time any subgroup discrimination, and thus eliminate model bias in the making. Additionally, this might be accomplished

through the implementation of DivExplorer and subgroup analysis during each evaluation phase with specific Pytorch model overrides.
It is therefore highly recommended to further extend the experiments conducted in this research by testing several more combinations of datasets and models, including less performing and more unbalanced datasets, with the aim of better highlighting the propagation of disparities between subgroups.

Chapter 6

Conclusions

6.1 Summary of Results

The incremental subgroup analysis carried out in this research showed, as a first step, how disparities between subgroups can emerge evolve and even out during the training of speech recognition models, both in the area of intent detection and ASR. Indeed, during the training of the models, many points were recognised where the divergences between the subgroups changed substantially, both positively and negatively compared to the baseline performance. It can be concluded from the results that with the training of several models of the similar architecture on the same dataset, very similar subgroup disparities occur.
This is not the case, however, if different architecture models are trained on the same dataset, which leads to different disparities between the groups. This highlighted how important it is to use a variety of architectures to highlight and better understand how they can influence subgroup bias and create divergences in performance. Finally, the use of DivExplorer was instrumental in enabling this research to study and monitor divergences between subgroups during model training.
Lastly, we can say that the results found in this research bode well for the implementation of an adaptive training framework that, by using DivExplorer and recalibrating the model itself during training, may be able to automatically correct disparities between subgroups in the bud so as to create models without any perceptible disparity in performance. This analysis is the framework adopted with this research wants to be part of the future ecosystem of methodologies that will make speech recognition models fairer and less biased.

Applicability to Contexts Beyond Speech Recognition

The focus of this study was entirely concentrated on the field of speech recognition, either ASR or intent detection. Naturally, the analysis of subgroups and their evolution through time certainly holds the potential to be extended into other fields as well. For instance, this may occur for an image classification context in which groups of classes with similar but not entirely equal characteristics were to be analysed. Indeed, using the same techniques that have been used in this research, one could incrementally monitor the appearance of bias and disparities in image recognition models as well and then provide the basis for corrective interventions. Therefore, the approach that has been studied and is presented in this thesis has potential that can be applied anywhere machine learning algorithms and artificial intelligences were used and not in domains strictly related to audio. This opens up the possibility for more cross-domain research and a deeper understanding of how subgroup performance evolves and propagates within artificial intelligence models.

6.2 Final Remarks

In conclusion, this research highlights how incremental subgroup analysis can bring to light performance disparities that emerge during the training and fine-tuning of speech recognition models. Although limited, the results obtained from the research clearly show that the disparities that emerge during training are clearly related to the intrinsic properties of the datasets and the models that use them. Among the macro themes emerging from this research we certainly have a better understanding of how subgroup disparities emerge spread and then converge during training and fine tuning of a variety of models.

A more specific analysis of the subgroups associated with the metadata of the datasets was then carried out, which led to a better understanding of how certain socio-linguistic variables such as gender and accent can influence the models' ability to recognise all speech genres. Lastly, theoretical insights were provided that, while remaining within the scope of analysis, will hopefully guide future studies for the development of speech recognition models that can mitigate subgroup disparities at birth.

Although this research has interesting results, it nevertheless wishes to acknowledge its limitations. These include the difficulty of generalising the results to different datasets and, in general, the partial and incomplete understanding of the learning dynamics of the models. May these issues, in

addition to overcoming the limitations mentioned above, be elements of future studies that can deepen these dynamics.

6.3 Acknowledgements

Firstly, I like to thank my professor Eliana Pastor and Ph.D Alkis Koudounas for their guidance during my internship and for giving me the opportunity to work on this thesis. A very special thanks to my best friend Ph.D Jacopo Franco who, other than always supporting me provided most of the computing power required for this dissertation model's training.
I would also like to express my deep gratitude to my family, who have always encouraged and supported me at every stage of my Turin excursion. To you above all: Dad, Mum and Chiara, I am deeply grateful Your presence in my life, even at a distance, was very meaningful to me. If I have been able to face this strange phase of my life with determination and hope, it is also thanks to you.
A special thanks goes to all my friends, new and old, who joined me during my time in Turin: Alex S., Alessandro, Vittorio, Geany, Natalia, Alex B., Marco, Riccardo, Giulio, Federica, Matteo, Sean, Alessio, Denis. Thank you for taking part in my life and in this period that will be difficult to forget.
Lastly, my deepest and most special gratitude goes to you: Davide, Raul and Massimo. Your presence in my life has been and continues to be fundamental. May I never stop thanking you for the good and happiness you have given me and which I hope I will still be able to share with you. To you I owe what I am today. Having you in my life makes me feel blessed, I could not have chosen better.

Bibliography

[1] A. Koudounas, E. Pastor, G. Attanasio, V. Mazzia, M. Giollo, T. Gueudre, E. Reale, L. Cagliero, S. Cumani, L. De Alfaro, E. Baralis, and D. Amberti, "Towards Comprehensive Subgroup Performance Analysis in Speech Models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1468–1480, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10430478/

[2] A. Koudounas, E. Pastor, G. Attanasio, L. De Alfaro, and E. Baralis, "Prioritizing Data Acquisition for end-to-end Speech Model Improvement," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 7000–7004. [Online]. Available: https://ieeexplore.ieee.org/document/10446326/

[3] A. Koudounas, E. Pastor, G. Attanasio, V. Mazzia, M. Giollo, T. Gueudre, L. Cagliero, L. De Alfaro, E. Baralis, and D. Amberti, "Exploring Subgroup Performance in End-to-End Speech Models," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rhodes Island, Greece: IEEE, Jun. 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10095284/

[4] E. Pastor, L. De Alfaro, and E. Baralis, "Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence," in *Proceedings of the 2021 International Conference on Management of Data*. Virtual Event China: ACM, Jun. 2021, pp. 1400–1412. [Online]. Available: https://dl.acm.org/doi/10.1145/3448016.3457284

[5] S. Feng, O. Kudina, B. M. Halpern, and O. Scharenborg, "Quantifying Bias in Automatic Speech Recognition," Apr. 2021.

[6] P. Dheram, M. Ramakrishnan, A. Raju, I.-F. Chen, B. King, K. Powell, M. Saboowala, K. Shetty, and A. Stolcke, "Toward Fairness in Speech Recognition: Discovery and mitiga-

tion of performance disparities," in *Interspeech 2022*. ISCA, Sep. 2022, pp. 1268–1272. [Online]. Available: https://www.isca-archive.org/interspeech$_2$022/$dheram22_interspeech.html$

[7] A. Koudounas, E. Pastor, L. d. Alfaro, E. Baralis *et al.*, "Mitigating subgroup disparities in speech models: A divergence-aware dual strategy," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 1–13, 2025.

[8] "The accent gap: How Amazon's and Google's smart speakers leave certain voices behind." [Online]. Available: https://www.washingtonpost.com/graphics/2018/business/alexa-does-not-understand-your-accent/

[9] L. Edmonds, "McDonald's is removing its AI drive-thru voice-ordering system from over 100 restaurants after its mishaps went viral," https://www.businessinsider.com/mcdonalds-ai-voice-order-technology-drive-thrus-2024-6.

[10] A. Saade, J. Dureau, D. Leroy, F. Caltagirone, A. Coucke, A. Ball, C. Doumouro, T. Lavril, A. Caulier, T. Bluche, T. Gisselbrecht, and M. Primet, "Spoken language understanding on the edge," in *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, 2019, pp. 57–61.

[11] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5754–5758.

[12] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[13] E. Pastor, A. Koudounas, G. Attanasio, D. Hovy, and E. Baralis, "Explaining speech classification models via word-level audio segments and paralinguistic features," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Y. Graham and M. Purver, Eds. St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 2221–2238. [Online]. Available: https://aclanthology.org/2024.eacl-long.136/

[14] A. Ananthaswamy, "A New Link to an Old Model Could Crack the Mystery of Deep Learning," Oct. 2021. [Online]. Avail-

able: https://www.quantamagazine.org/a-new-link-to-an-old-model-could-crack-the-mystery-of-deep-learning-20211011/

[15] A. Koudounas, F. Giobergia, E. Pastor, and E. Baralis, "A Contrastive Learning Approach to Mitigate Bias in Speech Models," in *Interspeech 2024*, Sep. 2024, pp. 827–831.

[16] I. Baldini, D. Wei, K. N. Ramamurthy, M. Yurochkin, and M. Singh, "Your fairness may vary: Pretrained language model fairness in toxic text classification," Apr. 2022.

[17] A. Koudounas, E. Pastor, E. Baralis *et al.*, "Assessing speech model performance: A subgroup perspective," in *SEBD 2024: 32nd Symposium on Advanced Database System*, vol. 3741. CEUR Workshop Proceedings, 2024, pp. 101–111.

[18] A. Koudounas and F. Giobergia, "Houston we have a Divergence: A Subgroup Performance Analysis of ASR Models," Mar. 2024.

[19] "Assessing and Mitigating Speech Model Biases via Pattern Mining."

[20] M. Kronis, *Harvesting Targeted Speech Data from Highly Expressive Found Spontaneous Speech by Learning Speaker Representations*, 2024.

[21] D.-W. Zhou, Z.-W. Cai, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Revisiting Class-Incremental Learning with Pre-Trained Models: Generalizability and Adaptivity are All You Need," Aug. 2024.

[22] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *Proceedings of the National Academy of Sciences*, vol. 117, no. 14, pp. 7684–7689, Apr. 2020.

[23] M. La Quatra, A. Koudounas, L. Vaiani, E. Baralis, L. Cagliero, P. Garza, and S. M. Siniscalchi, "Benchmarking representations for speech, music, and acoustic events," in *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, April 2024, pp. 505–509.

[24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[26] D. Resnik and M. Hosseini, "The ethics of using artificial intelligence in scientific research: new guidance needed for a new tool," *AI and Ethics*, pp. 1–23, 05 2024.

[27] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[28] A. Hinsvark, N. Delworth, M. D. Rio, Q. McNamara, J. Dong, R. Westerman, M. Huang, J. Palakapilly, J. Drexler, I. Pirkin, N. Bhandari, and M. Jette, "Accented Speech Recognition: A Survey," Jun. 2021.

[29] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed. [Online]. Available: https://web.stanford.edu/ jurafsky/slp3/

[30] D. Powers and Ailab, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *J. Mach. Learn. Technol*, vol. 2, pp. 2229–3981, Jan. 2011.

[31] C. Park, H. Kang, and T. Hain, "Character Error Rate Estimation for Automatic Speech Recognition of Short Utterances," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, Aug. 2024, pp. 131–135.

[32] admin, "Fluent Speech Commands: A dataset for spoken language understanding research," Apr. 2021. [Online]. Available: https://fluent.ai/fluent-speech-commands-a-dataset-for-spoken-language-understanding-research/

[33] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A Spoken Language Understanding Resource Package," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 7252–7262. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.588

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. South Brisbane, Queensland, Australia: IEEE, Apr. 2015, pp. 5206–5210. [Online]. Available: http://ieeexplore.ieee.org/document/7178964/

[35] A. Koudounas, M. La Quatra, L. Vaiani, L. Colomba, G. Attanasio, E. Pastor, L. Cagliero, and E. Baralis, "ITALIC: An Italian Intent Classification Dataset," 2023, pp. 2153–2157. [Online]. Available: https://www.isca-archive.org/interspeech$_2$023/$koudounas23_interspeech.html$

[36] J. FitzGerald, C. Hench, C. Peris, S. Mackie, K. Rottmann, A. Sanchez, A. Nash, L. Urbach, V. Kakarala, R. Singh, S. Ranganath, L. Crist, M. Britan, W. Leeuwis, G. Tur, and P. Natarajan, "MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages," Jun. 2022.

[37] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," Oct. 2020.

[38] "Facebook/wav2vec2-base · Hugging Face," https://huggingface.co/facebook/wav2vec2-base, Aug. 2024.

[39] "Facebook/wav2vec2-xls-r-300m · Hugging Face," https://huggingface.co/facebook/wav2vec2-xls-r-300m, Aug. 2024.

[40] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," Dec. 2021.

[41] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 993–1003. [Online]. Available: https://aclanthology.org/2021.acl-long.80

[42] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *ArXiv*, vol. abs/2012.03411, 2020.

[43] 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," Mar. 2020.

[44] H. Duan, J. Wang, K. Chen, and D. Lin, "Babel," dec 2024. [Online]. Available: https://service.tib.eu/ldmservice/dataset/babel

[45] J. Valk and T. Alumäe, "VoxLingua107: A Dataset for Spoken Language Recognition," Nov. 2020.

[46] J. Grosman, "Fine-tuned XLSR-53 large model for speech recognition in Italian," https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-italian, 2021.

[47] 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," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.

[48] "Facebook/hubert-base-ls960 · Hugging Face," https://huggingface.co/facebook/hubert-base-ls960, Aug. 2024.

[49] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," Jun. 2021.

[50] "Facebook/hubert-large-ll60k · Hugging Face," https://huggingface.co/facebook/hubert-large-ll60k, Jan. 2024.

[51] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673, https://github.com/facebookresearch/libri-light.

[52] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P. Mazare, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020, p. 7669–7673. [Online]. Available: http://dx.doi.org/10.1109/ICASSP40776.2020.9052942

[53] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et du Jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, no. 142, p. 547, 1901.

List of Figures

List of Tables

6.4 Appendix C: Code Listings

All the code for this thesis is available at: https://github.com/koudounasalkis/Speech-Incrementality

Build, to the future.