# Efficient Compression of Multitask Multilingual Speech Models

**Master of Science - Internship Report**

NAVER LABS Europe

at NAVER LABS Europe

PhD Track / Master M2 MVA
**Master of Science - Mathématiques, Vision, Apprentissage (MVA)**


| | |
|---|---|
| Scientific Advisor: | Prof. Dr. Emmanuel DUPOUX (PSL/Meta AI) |
| Internship Supervisor: | Dr. Marcely ZANON BOITO (NAVER LABS Europe)<br>Dr. Caroline BRUN (NAVER LABS Europe)<br>Dr. Vassilina NIKOULINA (NAVER LABS Europe) |
| Author: | Thomas PALMEIRA FERRAZ<br>Paris, France<br>thomas.palmeira@telecom-paris.fr |



Submission: 2nd October 2023

# Abstract

*Whisper* is a multitask and multilingual speech model covering 99 languages. It yields commendable automatic speech recognition (ASR) results in a subset of its covered languages, but the model still underperforms on a non-negligible number of under-represented languages, a problem exacerbated in smaller model versions. In this work, we examine its limitations, demonstrating the presence of speaker-related (gender, age) and model-related (resourcefulness and model size) bias. Despite that, we show that only model-related bias are amplified by quantization, impacting more low-resource languages and smaller models. Searching for a better compression approach, we propose *DistilWhisper*, an approach that is able to bridge the performance gap in ASR for these languages while retaining the advantages of multitask and multilingual capabilities. Our approach involves two key strategies: lightweight modular ASR fine-tuning of `whisper-small` using language-specific experts, and knowledge distillation from `whisper-large-v2`. This dual approach allows us to effectively boost ASR performance while keeping the robustness inherited from the multitask and multilingual pre-training. Results demonstrate that our approach is more effective than standard fine-tuning or LoRA adapters, boosting performance in the targeted languages for both in- and out-of-domain test sets, while introducing only a negligible parameter overhead at inference.

# Contents

1 Introduction . . . 1
1.1 Motivation . . . 1
1.2 Internship Objectives . . . 1
1.3 Contributions of this work . . . 2
1.4 About NAVER LABS Europe . . . 2
2 Background and Related Work . . . 3
2.1 State of the Art for Automatic Speech Recognition . . . 3
2.2 Domain Adaptation . . . 3
2.2.1 Low-rank Adapters (LoRA) . . . 4
2.3 Quantization . . . 5
2.4 Knowledge Distillation . . . 6
2.5 Datasets for Multilingual ASR . . . 6
2.5.1 CommonVoice 13.0 . . . 7
2.5.2 FLEURS . . . 7
2.6 The Whisper Model . . . 7
2.6.1 Overview . . . 8
2.6.2 Architecture . . . 9
2.6.3 Multitasking . . . 10
3 Bias Analysis on Quantized Speech Models . . . 12
3.1 Experimental Setup . . . 12
3.1.1 Dataset preparation . . . 12
3.1.2 Resourcefulness categorization . . . 12
3.2 Bias evaluation on Whisper . . . 13
3.3 Bias evaluation on quantized Whisper . . . 16
3.4 Summary of the main findings . . . 20
4 DistilWhisper . . . 21
4.1 Conditional Language-Specific Routing . . . 22
4.2 *DistilWhisper* approach . . . 23
4.3 *DistilWhisper* optimization . . . 23
4.3.1 Gate budget loss . . . 23
4.3.2 Knowledge Distillation . . . 24
4.3.3 Final Learning Objective . . . 24
5 Experiments and Results on DistilWhisper . . . 25
5.1 Experimental Setup . . . 25
5.1.1 Datasets . . . 25
5.1.2 Language Selection . . . 26
5.1.3 Models and Baselines . . . 27
5.1.4 Implementation details . . . 27

5.2 *DistilWhisper* versus other adaptation approaches . . . . . 28
5.3 Impact of knowledge distillation . . . . . 30
5.4 *DistilWhisper* Scalability . . . . . 30
5.5 Gate Activation Analysis . . . . . 31
5.6 Considerations on the Resourcefulness . . . . . 34
5.7 Effect of temperature and distillation loss . . . . . 34
5.8 Multi-domain training . . . . . 36
6 Conclusion . . . . . 39
6.1 Future Work . . . . . 39
References . . . . . IV
Appendices . . . . . VIII

# 1 Introduction

## 1.1 Motivation

Over the past three years, the field of Natural Language Processing (NLP) has been revolutionized by the introduction of large pre-trained models, often referred to as "foundation models." These models, both for text and speech, are trained on vast amounts of unlabeled data and can subsequently be fine-tuned for specific tasks using limited labeled data.

Multilingual foundation models have garnered significant attention due to their ability to handle hundreds of languages within a single model. However, they face a challenge known as the *curse of multilinguality*: in order to maintain high performance across all supported languages, these models require an increase in the number of parameters, leading to larger memory requirements and slower inference times. This can render the use of such models impractical in certain scenarios. To address this issue, research has been conducted on model compression techniques, although these methods may inadvertently exacerbate biases present in the model.

This internship focuses on OpenAI's Whisper, a family of multilingual multi-task speech models known for their impressive performance in speech recognition. These models exhibit robustness when transcribing speech recorded under various conditions, surpassing the capabilities of previous models.

However, there remain important questions to explore regarding Whisper and its multi-task learning approach. Although the model presents exceptional capability for transcribing and translating English, its performance in other languages indicates a decline in multilingual capabilities as the model size decreases. Additionally, we aim to investigate how this multilingual architecture handles biases related to different speakers, including gender, age, and accent. These questions drive our research to enhance the understanding of Whisper's capabilities and limitations.

## 1.2 Internship Objectives

This internship has three main objectives:

(1) Conduct a comprehensive analysis of bias within the Whisper model family, with a specific focus speaker-related (gender, age, accent) and model-related (model size, resourcefulness, similar languages) biases;

(2) Explore how light compression techniques, such as quantization, may either mitigate or exacerbate any identified biases within the Whisper models;

(3) Propose a better compression approach that effectively reduces any disparities found in the models.

## 1.3 Contributions of this work

This work offers two significant contributions. Firstly, it provides a comprehensive analysis of the biases present in the Whisper model and examines how quantization impacts these biases. Secondly, it introduces an alternative model compression method called *DistilWhisper*, which enhances the performance of smaller Whisper models. Additionally, all models and code developed in this research will be made available as open-source resources.

The structure of this report is as follows: Chapter 2 provides essential fundamentals and a comparison with related work to establish a foundational understanding. Chapter 3 details the experimental setup and results of the investigation into bias when quantizing Whisper. This investigation leads to the proposal of *DistilWhisper*, in Chapter 4, a novel parameter-efficient distillation approach that leverages small pre-trained models. Chapter 5 covers the validation of the proposed approach, as well as some interesting analysis. Finally, Chapter 6 summarizes the primary findings and conclusions of this work.

## 1.4 About NAVER LABS Europe

NAVER LABS is the R&D subsidiary of NAVER, Korea's leading internet company and the part of NAVER responsible for creating future technology. Its world-class researchers in Korea and Europe create new connections between people, machines, spaces and information by advancing technology in AI, robotics, autonomous driving, 3D/HD mapping and AR.

NAVER LABS Europe is the biggest industrial research lab in artificial intelligence in France and a hub of NAVER's global AI R&D Belt, a network of centers of excellence in Korea, Japan, Vietnam, USA & Europe. The scientists at NAVER LABS Europe conduct fundamental and applied research in machine learning (optimization, robotics), computer vision, natural language processing and UX and ethnography. The site is located in Grenoble, France.

# 2 Background and Related Work

## 2.1 State of the Art for Automatic Speech Recognition

Current ASR approaches primarily involve adapting pre-trained Transformer stacks (Vaswani et al., 2017), which are initially trained through self-supervised learning (SSL) on unlabeled audio data. These pre-trained models can vary in their use of pre-text tasks (e.g., wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), WavLM (Chen et al., 2022)) and the range of languages they cover (e.g., XLSR-53 (Conneau et al., 2021), XLS-R (Babu et al., 2022), MMS (Pratap et al., 2023), Google-USM (Y. Zhang et al., 2023)). This development of models has also seen the introduction of monolingual and multilingual SSL benchmarks. Examples of such benchmarks include SUPERB for English (Yang et al., 2021), LeBenchmark (Evain et al., 2021) for French, and ML-SUPERB (Shi et al., 2023), which covers 143 languages.

In contrast to this line of research, the Whisper model relies on weak supervision, meaning it is trained solely on weakly labeled data (without self-supervision). Nevertheless, with an ample amount of data, the Whisper model achieves competitive results when compared to monolingual (Gandhi et al., 2022; Radford et al., 2023) and multilingual (Pratap et al., 2023) SSL models. More details about Whisper can be found on Section 2.6. For broader ASR benchmarks, facilitating comparisons between SSL pre-training and multitasking weakly-supervised training, the ESB benchmark from HuggingFace (Gandhi et al., 2022) for English is an illustrative example.

## 2.2 Domain Adaptation

Domain adaptation consist in the process of adapting a pre-existing trained model to a new domain or task with minor weight adjustments, rather than retraining the entire model from scratch. In the past, this adaptation was primarily carried out through full fine-tuning, where all the model's weights were updated. In the case of Transformer-based models, it is also common to proceed adaptation choosing to update only specific layers, usually the final ones (Laskar et al., 2022).

More recently, the practice of domain adaptation has seen the emergence of Adapter-based techniques, initially proposed by Houlsby et al. (2019). Adapters are lightweight modules commonly used in both NLP and Speech to adapt pre-trained models to new tasks or domains. In speech-related tasks, Adapter-based fine-tuning has found applications in speech translation (Antonios et al., 2022; Gow-Smith et al., 2023; Le et al., 2021), domain adaptation (Thomas et al., 2022; Tomanek et al., 2021), and other

tasks. They have demonstrated comparable performance to standard fine-tuning while utilizing only a fraction of trainable parameters.

Furthermore, there are efforts to adapt Whisper models to specific tasks using LoRA adapters (e.g. Arabic dialect identification (Radhakrishnan et al., 2023), spoken language understanding (M. Wang et al., 2023), emotion recognition (Feng & Narayanan, 2023)). This technique is elaborated in Section 2.2.1. Additionally, some work involves full fine-tuning for task adaptation (e.g child spoken language understanding (Jain et al., 2023)).

In contrast to adapters and full fine-tuning, our work introduces gated Language-specific layers into the Whisper model and presents a parameter-efficient Knowledge Distillation approach. These innovations enhance the model's robustness to out-of-domain data.

### 2.2.1 Low-rank Adapters (LoRA)

Low-rank Adapter (LoRA) fine-tuning, as proposed by Hu et al. (2022), is a technique designed to reduce memory requirements for domain adaptation. This is achieved by introducing new trainable parameters into a pre-trained neural network while keeping the original pre-trained model weights fixed. These introduced parameters take the form of trainable rank decomposition matrices, and they are inserted between specific layers or blocks of the model. This approach significantly reduces the number of parameters that need to be fine-tuned when adapting the model for specific downstream tasks. For example, when fine-tuning a multilingual multi-task model for a single language and task, LoRA adapters help streamline the adaptation process.

The key assumption behind LoRA is that weight matrix updates in Transformer-based models exhibit a low "intrinsic rank" when undergoing full fine-tuning. This means that a pre-trained weight matrix, denoted as $W_0 \in \mathbb{R}^{d\times k}$, can be effectively represented using a low-rank matrix decomposition, denoted as $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}, A \in \mathbb{R}^{r\times k}$, and the rank $r \ll \min(d, k)$. Importantly, during LoRA fine-tuning, the $W_0$ part remains fixed (frozen) and does not receive gradient updates, while $A$ and $B$ become sets of trainable parameters.

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{2.1}$$

One significant advantage of this approach is that it allows for parallel computation during the forward pass. Specifically, the forward pass output $h$ can be efficiently computed

in parallel, and then the partial results are summed coordinate-wise, as presented in Equation 2.1.

## 2.3 Quantization

Quantization is a well-established technique in the field of Deep Learning, employed to increase the efficiency of neural networks. Historically, neural networks were often trained using low-precision numerical representations (Hubara et al., 2017). However, a recent trend, particularly in NLP, involves post-training quantization. This technique entails applying quantization to models after they have been trained with regular, higher precision. This approach has gained traction as it offers the dual benefits of reducing inference latency and model size.

Post-training quantization has found widespread use in various domains, including machine translation and language models (Bondarenko et al., 2021; Liang et al., 2021; Menghani, 2023; Wu et al., 2020). Quantized NLP models have yielded promising results, making it an appealing approach.

One of the most widely adopted techniques for post-training quantization in both NLP and speech communities is the **LLM.int8()** algorithm (Dettmers et al., 2022). This method implements quantization in the feed-forward and attention projection layers of the Transformer architecture. The method has two parts: vector-wise quantization and mixed precision decomposition. In the vector-wise quantization, it is determined conversion constants that allow for the recovery of original numbers from 8-bit to 16-bit floating-point representations. This enables matrix multiplication to be carried out in the lower 8-bit precision. Moreover, in the mixed precision decomposition, it identifies potential outliers that could be adversely impacted by reduced precision and then executes this part of the matrix multiplication in 16-bit precision.

While initially designed for decoder-only large language models (LLMs), this quantization method, along with its 4-bit variation (Dettmers & Zettlemoyer, 2023), has gained widespread adoption for various Transformer-based models. It has been made readily available in the Transformers library by Hugging Face (Wolf et al., 2020), contributing to its popularity. Additionally, it is becoming common to combine this quantization technique with domain adaptation methods. For instance, the QLoRA (Dettmers et al., 2023) method incorporates LoRA adapters on top of a quantized Transformer model.

## 2.4 Knowledge Distillation

Knowledge distillation (KD) has been initially proposed by Hinton et al. (2015) to distill knowledge from ensemble of models into a single model. Over time, KD has evolved to distill knowledge from a large teacher model into smaller student models (Mohammadshahi et al., 2022; Sanh et al., 2020; Shen et al., 2023). Knowledge distillation can be approached in two primary ways: representation matching or distribution matching. In this work, our focus is on distribution matching.

Traditional distribution matching knowledge distillation methods involves minimizing the Kullback–Leibler (KL) divergence between a teacher model and a student model. This is mathematically represented by Equation 2.2:

$$J_{\text{KL}} = D_{\text{KL}}(p \| q_\theta) = \mathbb{E}_{\mathbf{Y} \sim p}\left[\log \frac{p(\mathbf{Y})}{q_\theta(\mathbf{Y})}\right] \tag{2.2}$$

where $p$ is the teacher distribution, $q_\theta$ is the student distribution, and $\mathbf{Y}$ is sampled from the teacher distribution.

However, learning based on KL divergence at the sequence level can often lead to the student distribution becoming overly smooth, as it attempts to cover the entire support of the teacher distribution. This behavior arises due to the asymmetric nature of the KL divergence, a phenomenon sometimes referred to as the *mode-averaging problem*, as demonstrated by (Wen et al., 2023).

Recent research (Go et al., 2023; Wen et al., 2023) have shown that symmetric divergences, such as the Jensen-Shannon (JS) divergence, exhibit fewer borderline behaviors and tend to yield improved results in sequence-level distillation. Traditional JS divergence is expressed in Equation 2.3:

$$J_{\text{JS}} = D_{\text{JS}}(p \| q_\theta) = \frac{1}{2}\mathbb{E}_{\mathbf{Y} \sim p}\left[\log \tfrac{p(\mathbf{Y})}{m(\mathbf{Y})}\right] + \frac{1}{2}\mathbb{E}_{\mathbf{Y}' \sim q_\theta}\left[\log \tfrac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')}\right] \tag{2.3}$$

where $p$ is the teacher distribution, $q_\theta$ is the student distribution, $\mathbf{Y}$ and $\mathbf{Y}'$ are sampled from the teacher's and student's distributions and compared with their average $m(\cdot) = \frac{1}{2}p(\cdot) + \frac{1}{2}q_\theta(\cdot)$.

## 2.5 Datasets for Multilingual ASR

Here we present two widely used massively-multilingual datasets that will be used in this work: CommonVoice 13.0 and FLEURS.

### 2.5.1 CommonVoice 13.0

The CommonVoice 13.0 (CV-13) corpus (Ardila et al., 2020), represents the latest iteration of a massively multilingual collection of transcribed speech. It serves as a valuable resource for research and development in the field of speech technology. While primarily designed for Automatic Speech Recognition (ASR) applications, this dataset also finds utility in other domains, such as language identification. The utterances comprising this dataset are sourced from Wikipedia articles and supplemented with utterances contributed by language communities. These are subsequently narrated by contributors through Mozilla's website or iPhone app. To ensure data quality, contributions undergo validation by other volunteers, with only validated data being incorporated into the train, validation, and test subsets splits of the dataset. As of the current version, the dataset encompasses a rich tapestry of 110 languages, though the number of utterances per language varies significantly.

### 2.5.2 FLEURS

The FLEURS (Conneau et al., 2023) is an n-way parallel speech dataset in 102 languages built on top of the machine translation FLoRes-101 benchmark (Goyal et al., 2022), with approximately 12 hours of speech supervision per language. It was meant for few-shot learning on a variety of speech tasks, including Automatic Speech Recognition, Speech Language Identification, Speech Translation and Retrieval. The creation of this dataset involved the recording of all the publicly available sentences from FLoRes-101 (from dev and devtest split subsets). Each sentence was recorded by three paid native-speaker experts per language. Subsequently, these spoken sentences underwent a thorough evaluation by paid evaluators to ensure the overall quality and accuracy of the recorded content. The dataset is unbalanced as not all the sentences were validated, but most part of the languages have between 2400 and 3300 utterances on the train split, with an average 12 seconds per audio sample.

## 2.6 The Whisper Model

In this section we present Whisper (Radford et al., 2023), the base model for the studies conducted in this work.

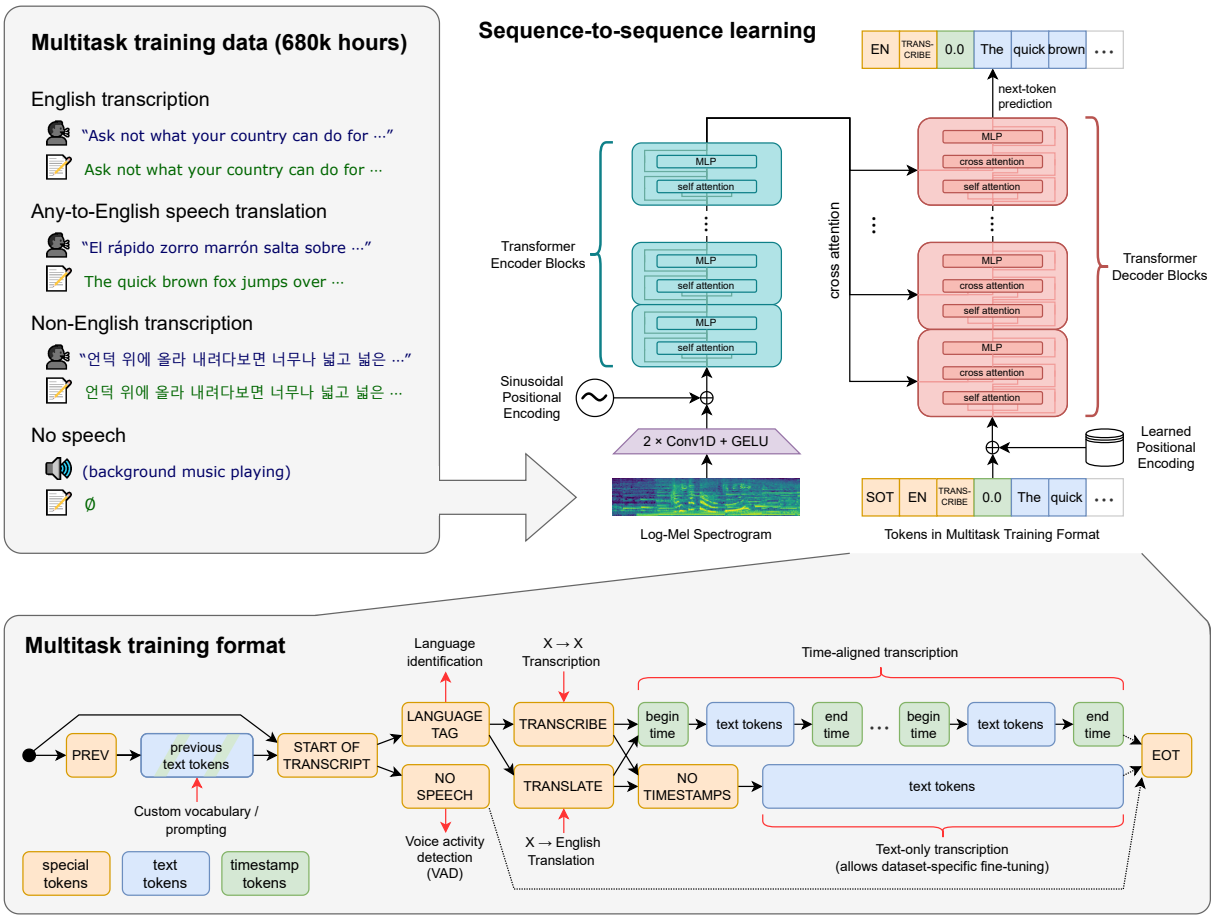


**Figure 1** The Whisper model architecture (Source: Radford et al. (2023))

### 2.6.1 Overview

Whisper is designed to serve as a versatile end-to-end Automatic Speech Recognition (ASR) model suitable for a wide range of applications and languages. When it comes to ASR, previous research has predominantly focused on two key approaches: large-scale Unsupervised Learning (Y. Wang et al., 2022) and Supervised Learning as discussed in Section 2.1.

In the case of large-scale Unsupervised Learning, models benefit from training on vast, low-cost, and unlabeled datasets, which helps in building a high-quality encoding component. However, these models generate output that is not directly usable for ASR applications and requires further fine-tuning. On the other hand, Supervised Learning approaches utilize pretrained models that can be directly used for ASR tasks. However, they often struggle to generalize when faced with shifts in the data distribution, primarily due to the limited size of the datasets they were originally trained on. Additionally, creating large-scale human labeled datasets for these models can be prohibitively expensive.

Whisper takes a unique approach by introducing Weakly Supervised Learning, striking a balance between data quality and quantity. The Whisper training dataset is curated by collecting pairs of audio and corresponding transcripts from the internet (mainly YouTube videos). After some minimal processing, that included employing language identification with the model proposed by Valk and Alumäe (2021), this dataset comprises a substantial $680,000$ hours of highly diverse audio content. Notably, it encompasses 96 languages besides English, with approximately 17.2% of the dataset consisting of audio and transcript pairs in the same language (ASR). Additionally, around 18.4% of the pairs have English-translated transcripts.

This unique approach provides Whisper with several advantages. Firstly, the Whisper encoder benefits from the rich and diverse dataset, making it perform exceptionally well, similar to Unsupervised settings. Secondly, Whisper is trained with relatively clean labels, allowing it to be used in a Zero-Shot manner without the need for extensive fine-tuning.

### 2.6.2 Architecture

The architecture of Whisper consists of the original Transformer architecture (Vaswani et al., 2017) preceded by dimension reduction layer called stem. The architecture is visually depicted in Figure 1.

*Stem*

The stem comprises a pair of 1-dimensional Convolution Layers, each accompanied by GELU activations. Both convolution layers employ filters of size $3$ and produce $d$ output channels. The value of $d$ varies across different sizes of the Whisper architectures. The first convolution layer operates with a stride of $1$, while the second employs a stride of $2$ (effectively reducing the length of the input sequence by half). Consequently, the output of the stem consists of a sequence of $1500$ elements, each with dimension $d$. As the self-attention layers in a Transformer exhibit quadratic complexity concerning the sequence length, for a fixed hidden representation size of $d$, the stem significantly reduces the computational complexity by a factor of $4$.

*Transformer*

In their work, Radford et al. (2023) primarily highlights the impact of scaled Weak Supervision on ASR system performance, with less emphasis on architectural modifications. The base architecture employed for Whisper is the encoder-decoder Trans-

former, which is renowned for its scalability and reliability in several sequence-to-sequence tasks.

However, the Whisper Transformer does introduce a few key modifications compared to the original Transformer architecture. Sinusoidal encodings are added to the input representations of the encoder, while the positional encodings in the decoder are learned. Additionally, GELU activation functions are used instead of ReLU, and these activations are applied following the residual blocks. Moreover, a normalization layer is included in the encoder's output. Furthermore, Whisper offers a range of five different architecture sizes, as detailed in Table 1. These varying sizes cater to different requirements and performance needs, allowing for flexibility in ASR tasks.

| Model | Layer ($L$) | Width ($d$) | Parameters |
|---|---|---|---|
| Tiny | 4 | 384 | 39M |
| Base | 6 | 512 | 74M |
| Small | 12 | 768 | 244M |
| Medium | 24 | 1024 | 769M |
| Large | 32 | 1280 | 1550M |

**Table 1** Architectural specifications for the Whisper model family. $L$ denotes the number of layers per block, indicating that, for example, the tiny model with $L = 4$ consists of 4 transformer layers in the encoder and 4 in the decoder.

*Tokenization*

To tokenize transcripts, the Whisper model employs the BPE (Byte Pair Encoding) tokenizer originally introduced in GPT-2 by Radford et al. (2019). When dealing with languages other than English, the tokenizer is adapted by refining it until the vocabulary size matches that of English.

### 2.6.3 Multitasking

Whisper is trained and operates as a multitask model, capable of handling various sub-tasks within a single end-to-end architecture. These sub-tasks encompass Voice Activity Detection, Language Identification, Text Alignment, Transcription, Translation, and more. To delineate each task and the expected format of the subsequent predictions, specific tokens are employed, as delineated in Table 2. These tokens are positioned at the start of the output sequence, providing task context (see Figure 1). Token generation follows an auto-regressive process, reliant on prior tokens. For ex-

ample, when the detected language is French, the model computes the likelihood of token $w$ at position $k'$, as illustrated in Equation 2.4:

$$P(w_{k'} = w | \ldots, \texttt{<|fr|>}, \texttt{|transcribe|}, \ldots, w_{k'-1}, X) \tag{2.4}$$

Consequently, the generated tokens will probably only belong to the French vocabulary as they have higher conditional probabilities compared to ones belonging to other languages.

| **Tasks** | **Tokens** |
|---|---|
| Language Identification | `<\|LANGUAGE\|>`<br>e.g. `<\|en\|>`, `<\|gl\|>`, `<\|fr\|>`, `<\|fa\|>`, etc. |
| Voice Activity Detection | `<\|nospeech\|>` |
| Transcribe | `<\|transcribe\|>` |
| Translate | `<\|translate\|>` |
| Alignment | `<\|notimestamps\|>` |

**Table 2** Subset of special tokens associated with Whisper's multitasks. For Language Identification, each language is specified with a token, and a single token is added to the sequence. This token is required. For Voice Activity Detection, only when the audio does not contain clear speech that its corresponding token is present in the output. The tasks Transcribe and Translate are mutually exclusive, but one of them is required.

Additionally, certain special tokens can be predefined to simplify predictions. In our work, we specifically enforce transcription and language tokens, thereby eliminating dependency on Language Identification quality for under-represented languages. Tasks not pertinent to our study are also disregarded.

# 3 Bias Analysis on Quantized Speech Models

In this chapter, we aim at addressing the two first objective of the internship: understand the bias presented on Whisper models, and investigate how these are impacted by the employment of quantization.

## 3.1 Experimental Setup

### 3.1.1 Dataset preparation

In our research, we employed the two widely recognized datasets described in Section 2.5: FLEURS and Common Voice 13.0 (CV-13). These datasets provide valuable speaker-related information, including gender, language group (in the case of FLEURS), accent (exclusive to CV-13), and age (exclusive to CV-13).

Building upon the information available in FLEURS, we curated a gender-balanced benchmark, which we refer to as **Balanced-FLEURS**. The primary goal here was to mitigate the influence of confusion variables such as sentence complexity and gender imbalance (where certain languages exhibit a higher percentage of speakers from one gender). To achieve this, we mixture the train, validation, and test sets of FLEURS, meticulously filtering them to ensure that each sentence was narrated by both a male and a female speaker. Meanwhile, we also ran a Voice Activity Detection model on the dataset, as we encountered a notable number of empty audio files in Spanish, Norwegian, and Malay[1]. We include in the experiments only the languages in which we were able to find at least 200 utterances.

In addition to **Balanced-FLEURS**, we made use of the Common Voice 13.0 dataset, specifically its validation set. In this case, we leveraged gender and age information. While we attempted to incorporate accent information in our study as well, we encountered challenges in aggregating a sufficiently large dataset, even after merging the train, test, and validation splits. Consequently, we do not report our results with respect to accents.

### 3.1.2 Resourcefulness categorization

In the course of our experiments, we have introduced a resourcefulness classification system specifically tailored to weakly-supervised speech models, with a primary focus

[1] We have reported this issue to the Google Team via HuggingFace, listing all problematic files. The corresponding issue can be found here:
https://huggingface.co/datasets/google/fleurs/discussions/16#6442a217f8b647fa4f50c489

on the transcription task (ASR). This categorization is designed to group languages based on the amount of training data used in the model pre-training. The classification involves clustering languages into categories with similar amounts of training data, and the intervals used for this classification can be found in Table 3.

| **Resourcefulness** | **ASR Training data (h)** |
|---|---|
| Super High-Resource | $\geq 5000$ |
| High-resource | [1000, 5000) |
| Mid-to-high-resource | [500, 1000) |
| Low-to-mid-resource | [100, 500) |
| Low-resource | [10, 100) |
| Extremely Low-Resource | (0, 10) |

**Table 3** Proposed Language resourcefulness categorization for Weakly-supervised ASR models

It is worth noting that our proposed classification system has a limitation in the context of Whisper. Specifically, it does not account the volume of training data available for the speech translation task. While this data does not directly impact the quality of generated text data for a language (since in Whisper, translation data available is to English only), it does play a role in enhancing the model's speech encoding capabilities.

### 3.2 Bias evaluation on Whisper

In this section, we present preliminary experiments conducted on the Whisper model. Our aim here is to investigate whether bias exists in the original versions of Whisper. To achieve this, we compare Whisper's performance on the validation split of CV-13 and on Balanced-FLEURS. Our analysis involves an aggregate approach, where we average the metrics across languages.

Figures 2 (Balanced-FLEURS) and 3 (CV-13) showcase the Word Error Rate (WER) performance across the languages covered in the two datasets for `whisper-large-v2`. These results reveal a clear correlation between performance and resourcefulness, with lower resource languages (Low and Extremely Low-Resource) consistently exhibiting poorest performance. Naturally, the impact varies among languages, possibly due to their complexity or the amount of training data available for closely-related languages. These findings collectively suggest a bias linked to resourcefulness.

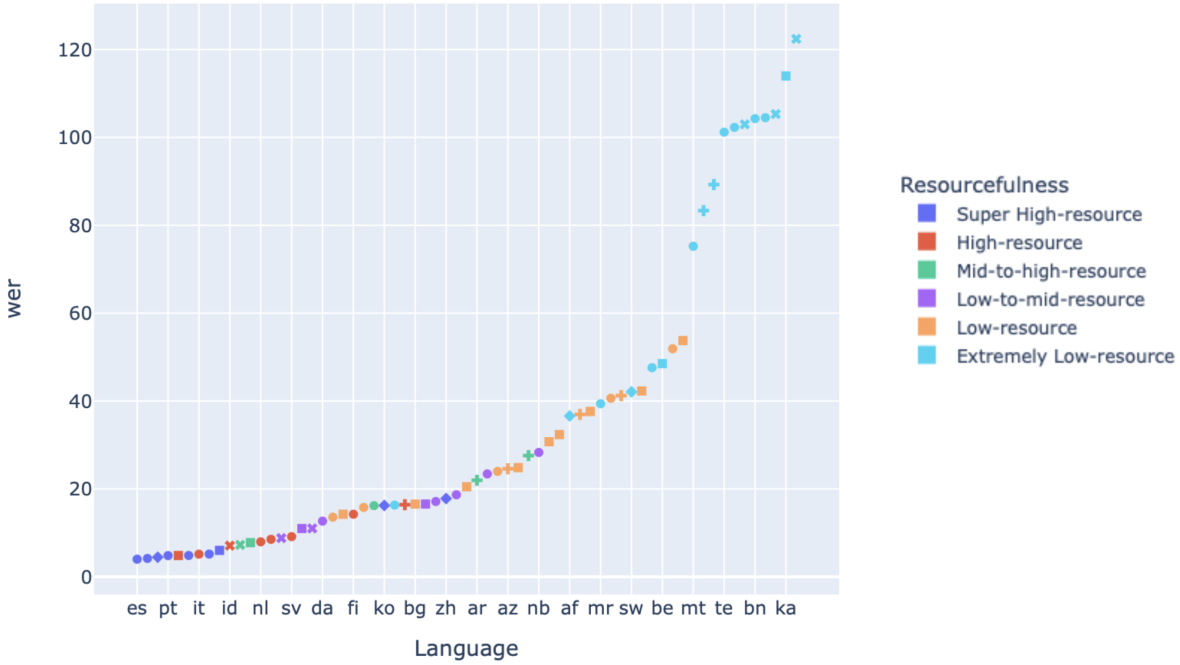


**Figure 2** Performance across languages on whisper-large-v2 on Balanced-FLEURS. Languages are ranked on x-axis based its performance.

Figure 4 illustrates the average relative difference between male and female speakers for Balanced-FLEURS on `whisper-large-v2`. This metric, already employed is previous similar study by Boito et al. (2022), is relevant here as the sentences are consistently the same across genders. Meanwhile, Figure 5 displays the absolute difference (following Costa-jussà et al. (2022)) in WER between male and female speakers on CV-13. In both cases, the results show varying degrees of gender bias across different languages. Remarkably, these biases are consistent across the different datasets, implying that each language possesses its unique bias, likely attributed to the quality and diversity of its training data. While the model does exhibit gender bias, it is essential to note that, for the most part, this bias remains within a maximum average WER difference of 3 for the majority of languages (in the case of CV-13).

Figure 6 extends the analysis by presenting WER performance across different languages on Balanced-FLEURS, mirroring Figure 2. However, this time, we consider all available model sizes within the Whisper family. Languages are ranked by resourcefulness. These results unveil two significant findings: (i) the performance trend aligns across nearly all languages, suggesting a consistent ranking of languages based on performance across all models; and (ii) notably, a clear correlation emerges between smaller model sizes and reduced performance, with the model curves closely overlapping. This phenomenon likely stems from the *curse of multilinguality*, wherein less resourceful languages exhibit larger performance disparities among model sizes. Addi-

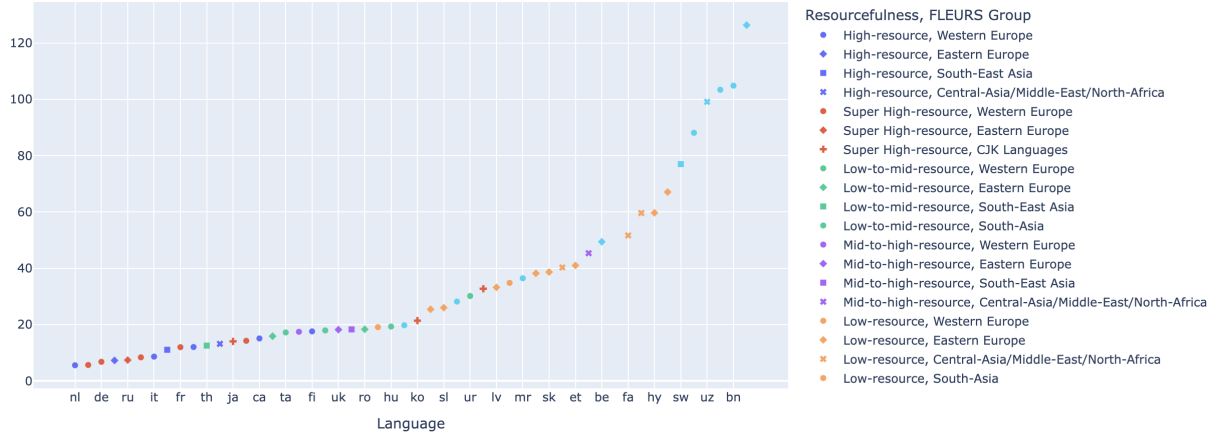


**Figure 3** Performance across languages on whisper-large-v2 on CV-13. Languages are ranked on x-axis based its performance.

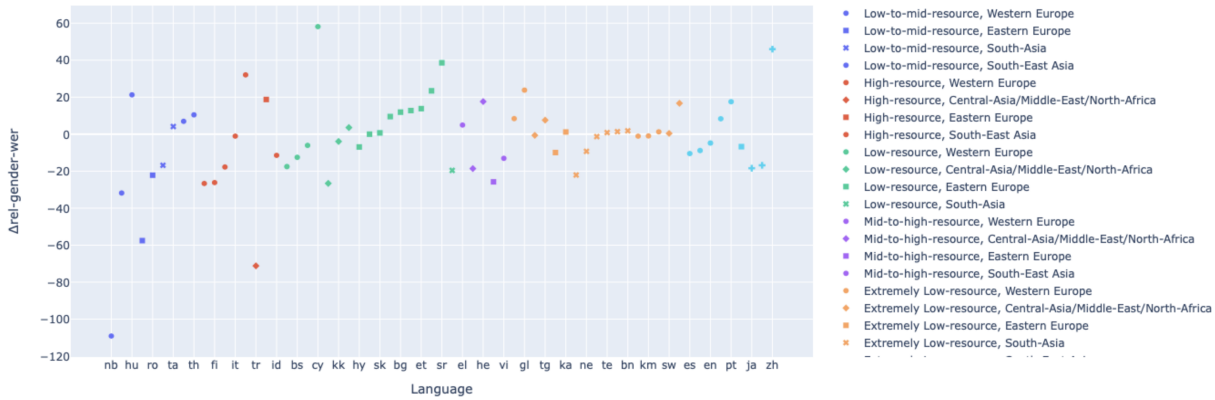


**Figure 4** Average relative WER difference between male and female voice for Balanced-FLEURS. Languages are ranked on x-axis based its relative difference and resourcefulness.

tionally, it's worth noting the differences between large and large-v2 models. Although both models share the same size, the former benefits from more extensive training, additional optimization steps, and data augmentation techniques. Finally, these findings collectively shed light on bias associated with architecture size, despite models being trained with the same dataset.

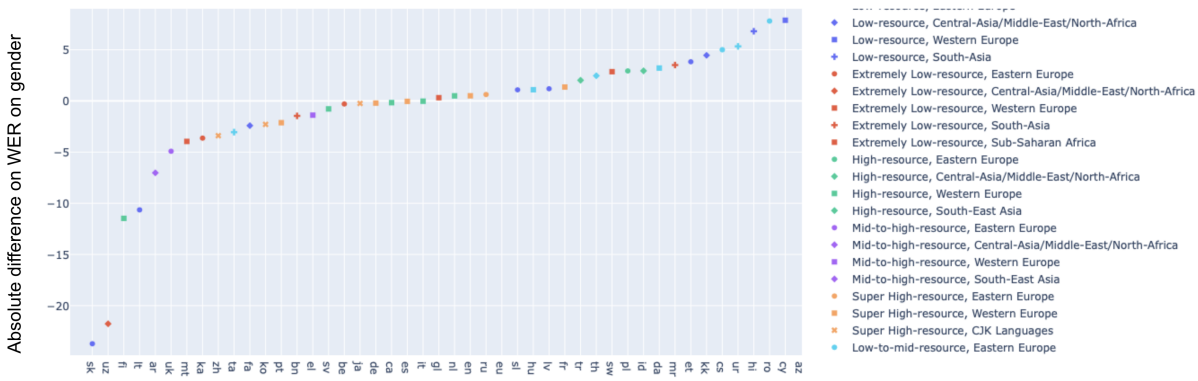


**Figure 5** Absolute WER difference between male and female voice for CV-13. Languages are ranked on x-axis based its absolute difference.

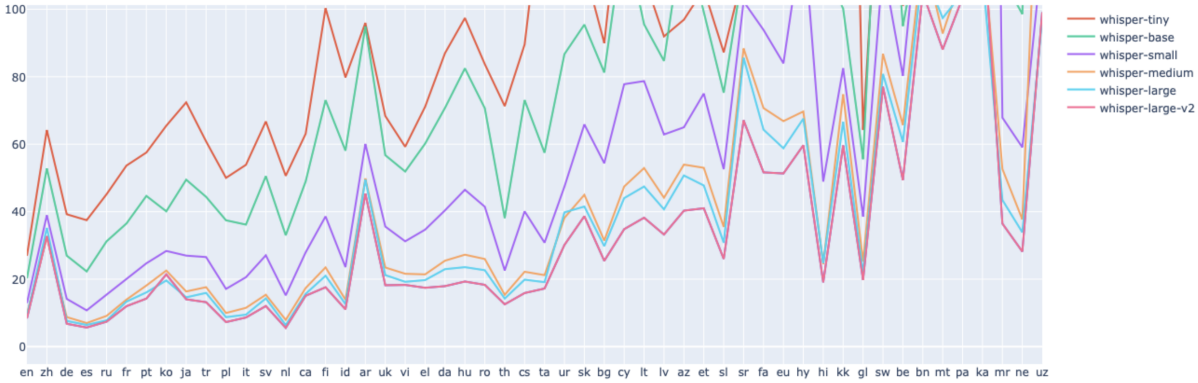


**Figure 6** Performance across languages and across different whisper sizes on Balanced-FLEURS. Languages are ranked on x-axis based its resourcefulness.

### 3.3 Bias evaluation on quantized Whisper

Now, we delve into the quantized version of Whisper. In this set of experiments, we apply the `LLM.int8()` method (Dettmers et al., 2022) (described in Section 2.3) to Whisper. The primary objective of this study is to investigate whether the biases observed in the original Whisper model persist, diminish, or intensify after quantization. In essence, we seek to understand what model features may be forgotten due to quantization.

In contrast to the previous section, our analysis here adopts a sentence-level approach. We compare the model's performance on individual sentences before and after quantization. To ensure a fair evaluation, we exclude sentences with initial Word Error Rate (WER) values greater than or equal to 100. For this sentence-level analysis, we create histograms based on the absolute difference in WER before and after compression. We categorize sentences into three groups: those that worsened (WER increased by

more than 5), those that remained similar (WER difference less than 5), and those that improved (WER reduced by more than 5).

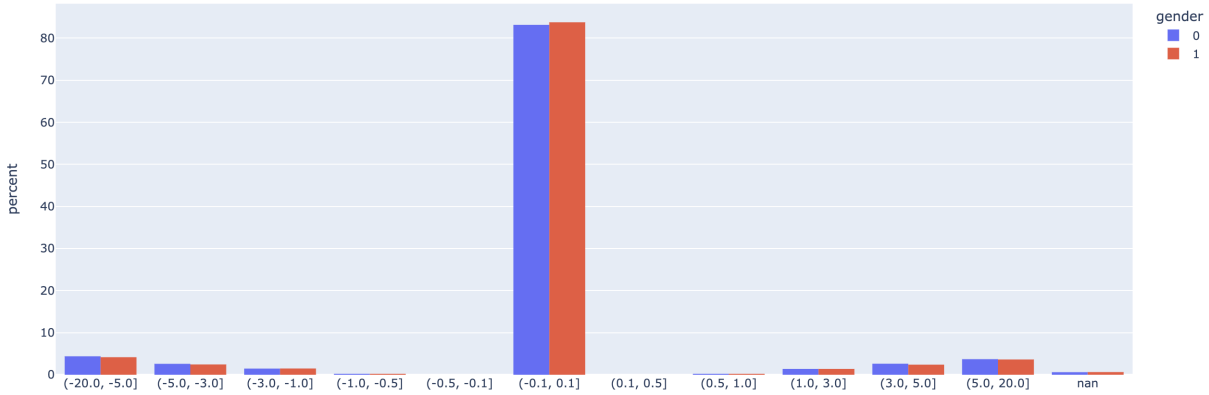


**Figure 7** Histogram of performance degradation by quantization per gender on Balanced-FLEURS

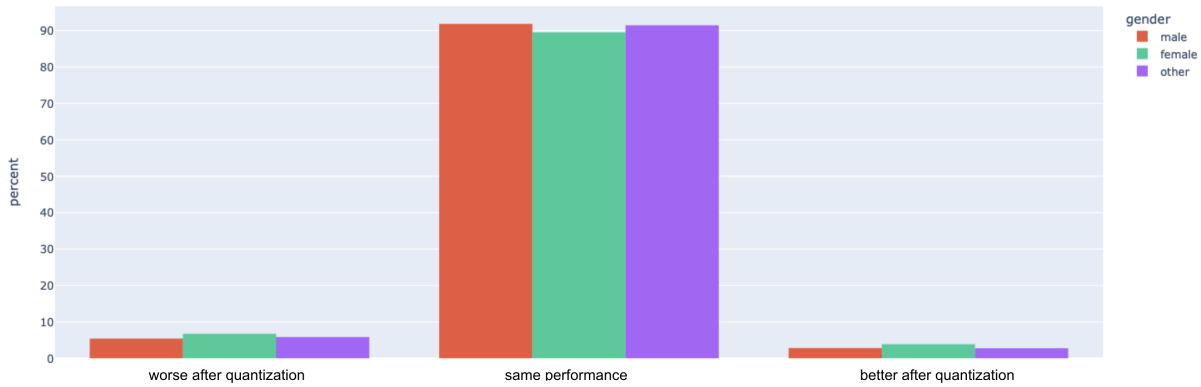


**Figure 8** Histogram of performance degradation by quantization per gender on CV-13

Figures 7 (Balanced-FLEURS) and 8 (CV-13) present histograms categorized by gender for the `whisper-large-v2` model. Figure 3 displays histograms categorized by age group for CV-13. The data clearly indicates that quantization equally impacts all genders and age groups, implying that gender and age biases are kept unchanged after quantization.

In figures 10 (Balanced-FLEURS) and 11 (CV-13), we illustrate histograms categorized by language resourcefulness for `whisper-large-v2`. Here, a distinct pattern emerges: lower-resource languages are more significantly affected by quantization. While almost all sentences in super high-resource languages maintain their performance, approximately 25% of sentences in extremely low-resource languages are impacted (in the case of Balanced-FLEURS). Consequently, quantization amplifies the resourcefulness bias.

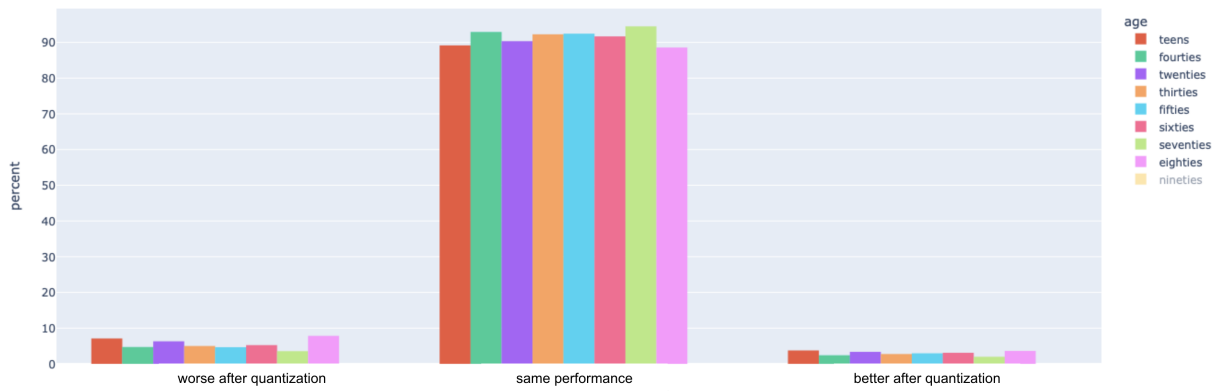


**Figure 9** Histogram of performance degradation by quantization per age group on CV-13

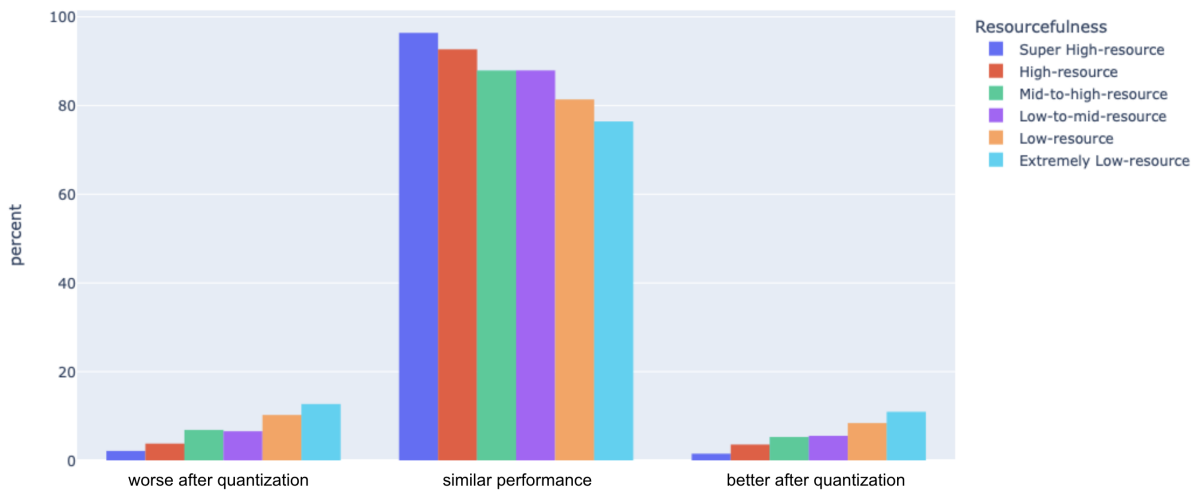


**Figure 10** Histogram of performance degradation by quantization per resourcefulness group on Balanced-FLEURS

Lastly, in figure 12 (Balanced-FLEURS) and **??** (CV-13), we present histograms considering all available model sizes within the Whisper family, grouped by model size. The results highlight significant differences in how quantization affects models of varying sizes. While a small proportion of sentences are impacted for `whisper-large-v2`, there is a striking contrast, with almost half of the sentences affected in the case of `whisper-tiny`. This highlights that the bias related to architecture size is significantly amplified by quantization.

This last finding indicates that smaller models are generally more susceptible to the effects of quantization. This observation is particularly concerning as many parameter-efficient domain adaptation methods in use today in NLP and Speech involve applying quantization first, without considering the model size. This calls for practitioners to

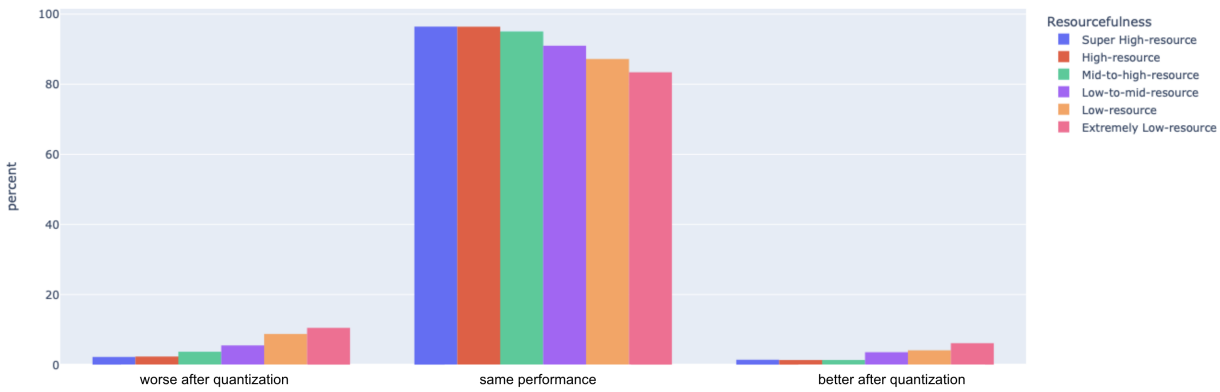


**Figure 11** Histogram of performance degradation by quantization per resourcefulness group on CV-13

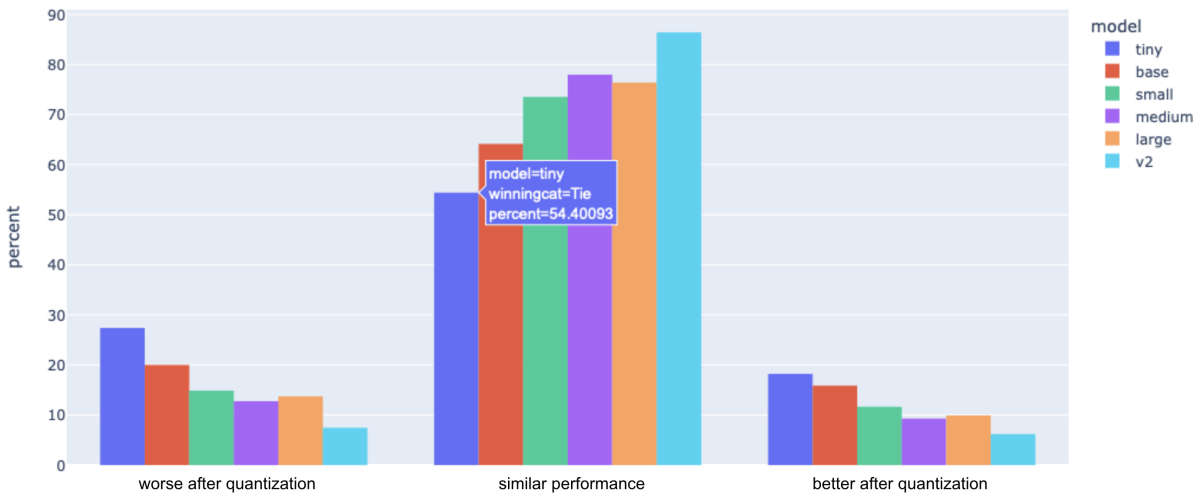


**Figure 12** Histogram of performance degradation by quantization per model size on Balanced-FLEURS

exercise caution when adapting pre-trained models to avoid the addition of unintended bias.

### 3.4 Summary of the main findings

Here we present the key takeaways from this chapter. First, Whisper exhibits certain speaker-related biases, such as gender and age. These biases are kept unchanged after applying quantization to the model.

On the other hand, biases associated with the model itself (model-related bias), including language resourcefulness and architecture size, are amplified by quantization. Overall, Low-resource languages are the most adversely affected by quantization. Moreover, there is a clear pattern on the architecture size, with smaller models experiencing more significant performance degradation compared to larger ones. This is concerning as current parameter-efficient approaches (such as QLoRA presented on Section 2.3) mostly apply quantization first, regardless of the model size.

This presents a significant challenge: Can we enhance the performance of smaller models for languages where they currently perform poorly, even though the best model performs well? We aim at searching an alternative to quantization to reduce the model size.

# 4 DistilWhisper

One prominent observation is the significant Automatic Speech Recognition (ASR) performance gap between the `whisper-large-v2` model and its counterparts of smaller sizes, especially when applied to a diverse set of languages. This gap in performance is noticeable across a wide spectrum of languages, that include the low-resource ones, but also many mid- and high-resource languages. As our earlier analysis, outlined in Chapter 3, revealed, the "lower" resource languages are also the most affected by lightweight compression techniques.

This phenomenon is often referred to as the *curse of multilinguality* (as discussed in related works by Arivazhagan et al. (2019), Conneau et al. (2020), and Goyal et al. (2021)). It stems from the inherent challenge that arises when attempting to cover an extensive array of languages within a single model - the performance inevitably suffers unless the model is significantly scaled up. This leads us to the central question that has motivated our research: Can we improve the performance of smaller models for languages in which they currently perform poorly, but the best model performs well?

A common approach to address this challenge of achieving efficient inference could be distilling knowledge from a larger multilingual teacher model into a smaller pre-existing one, as highlighted in prior works such as the ones done by Sanh et al. (2020) and Mohammadshahi et al. (2022). However, when it comes to applying such knowledge distillation (KD) to `whisper-large-v2`, which represents the best and largest Whisper model, we face a significant hurdle. This is because we need access to information that is not readily available, such as comprehensive training data spanning all tasks and languages, and its original learning objective, in order to maintain the original model's robustness.

Recent research findings, exemplified by works like Pfeiffer et al. (2022) and Pratap et al. (2023), have demonstrated an alternative solution to the *curse of multilinguality*. This approach involves equipping moderately sized models with language-specific (LS) modules. This sparse architectural design permits the extension of model parameters through additional modules as more languages are incorporated into the model. Consequently, it ensures consistent performance across languages without incurring substantial additional computational costs during inference.

In light of the overarching goal to enhance model performance for various languages within the constraints of limited model capacity, our work introduces the *DistilWhisper* approach. We incorporate conditional language-specific routing (CLSR) modules, as described by B. Zhang et al. (2021), into a smaller version of Whisper. We then opti-

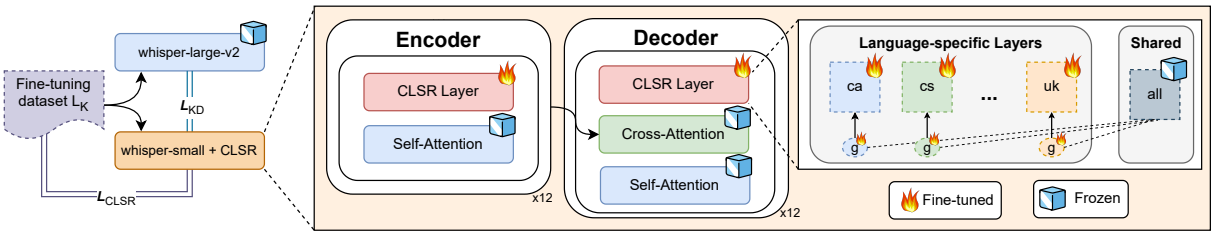


**Figure 13** The *DistilWhisper* optimization approach (left), and its architecture (right). The feed-forward is replaced by a CLSR module, where the LS gates (g) learn to alternate between the pre-trained frozen multilingual representation and the LS layer.

mize these modules jointly through ASR fine-tuning and knowledge distillation from a larger Whisper model (`whisper-large-v2`). For a visual representation of our architecture, please refer to Figure 13, and in the subsequent sections, we delve into the key components of our approach.

Following, in this chapter, we detail the elements that make up our approach. Then, in the next chapter (Chapter 5), we will present how we validate this approach and its results following the *DistilWhisper* approach presented here.

## 4.1 Conditional Language-Specific Routing

We extend Conditional Language-Specific Routing (CLSR) modules proposed by B. Zhang et al. (2021), and commonly used in Multilingual Neural Machine Translation, for the first time to the speech domain. This module, which introduces sparsity to the Transformer architecture, learns a hard binary gate $g(\cdot)$ for each input token by using its hidden embedding $z^l$. These decisions enable a layer to selectively guide information through either a LS path denoted as $h^{lang}$ or a shared path referred to as $h^{shared}$, as in Eq. 4.1:

$$\texttt{CLSR}(z^l) = g(z^l) \cdot h^{lang}(z^l) + (1 - g(z^l)) \cdot h^{shared}(z^l). \tag{4.1}$$

In contrast to the original CLSR, in this work we use language-specific gates as shown in Figure 13, instead of sharing them across languages. This allows us to train language-specific components individually (i.e. in parallel), and then only load the relevant modules at inference. Moreover, our approach also differs from the original CLSR by the positioning: supported by previous work (Pfeiffer et al., 2022; B. Zhang et al., 2021), we limit CLSR to the feed-forward network (correspondent to the feature domain of the Transformer architecture), which we also replace entirely by the CLSR module, reducing the increment in the number of parameters.

Following the proposal from B. Zhang et al. (2021), each gate $g(.)$ is made by a two-layer bottleneck network, which is summed to a increasing zero-mean Gaussian noise during training to discretize it:

$$g(z^l) = \sigma(G(z^l) + \alpha(t) \cdot \mathcal{N}(0,1)), \tag{4.2}$$

$$\text{with } G(z^l) = \texttt{ReLU}(z^l W_1 + w_2), \tag{4.3}$$

where $\sigma(\cdot)$ is the logistic-sigmoid function, and $W_1$ and $w_2$ are trainable parameters. $\alpha$ is linearly increased along with training steps $t$. At inference time, we adopt hard gating:

$$g(z^l) = \delta(G(z^l) \geq 0), \tag{4.4}$$

where $\delta(\cdot)$ is a Dirac measure.

## 4.2 *DistilWhisper* approach

Figure 13 presents our proposed *DistilWhisper* architecture. Our student is enriched with CLSR modules at each feed-forward for each language. These all experts in each CLSR layer are equally initialized from the frozen weights of the corresponding feed-forward layer. At training time, for each language the model updates only the corresponding language-specific experts and gates. At inference time, the model loads the shared layers (multilingual) and the Language-Specific experts and gates for the languages of interest, resulting in a limited parameter overhead. We highlight that the use of CLSR modules brings more flexibility to our architecture when compared to adapters, as it allows for routing at the token-level. This makes this approach more capable of leveraging pre-existing knowledge (shared frozen module), activating the Language-Specific path only when this is likely to increase performance.

## 4.3 *DistilWhisper* optimization

The optimization of our *DistilWhisper* architecture consist of a standard cross-entropy loss, along with two new elements: gate budget loss, and knowledge distillation. Following we detail these new elements.

### 4.3.1 Gate budget loss

Following B. Zhang et al. (2021), when learning CLSR module parameters, in addition to standard cross-entropy loss $\mathcal{L}_{\text{CE}}$, we optimize a gate budget loss $\mathcal{L}_{\text{g}}$ to balance

models' usage of language-specific and shared modules. It relies on the gate $g(.)$ activation values for a pair (audio, text) $(X, Y)$ in a batch $\mathcal{B}$, which is expressed by:

$$\mathcal{G}_{(X,Y)} = \sum_{x \in X} \sum_{m \in \mathcal{M}_{\text{enc}}} g_m(x) + \sum_{y \in Y} \sum_{m \in \mathcal{M}_{\text{dec}}} g_m(y) \tag{4.5}$$

where $\mathcal{M}_{\text{enc}}$ and $\mathcal{M}_{\text{dec}}$ are respectively the sets of encoders and decoders layers, and $g_m(.) = 1$ when LS expert is selected in the layer $m$, or $g_m(.) = 0$ otherwise. The average of this gate usage, representing the amount of language-specific experts used for the model in the batch, is constrained to a budget $b$. So the final gate budget loss is expressed by:

$$\mathcal{L}_{\text{g}} = \left| \frac{\sum_{(X,Y) \in \mathcal{B}} \mathcal{G}_{(X,Y)}}{\sum_{(X,Y) \in \mathcal{B}} (|X||\mathcal{M}_{\text{enc}}| + |Y||\mathcal{M}_{\text{dec}}|)} - b \right| \tag{4.6}$$

For regularization, also it is used a skip gate probability ($s$), that randomly choose a proportion $s$ of the gates to be closed (use only shared part) during training.

### 4.3.2 Knowledge Distillation

For Knowledge Distillation (KD), following recent research (Go et al., 2023; Wen et al., 2023), we employ Jensen–Shannon divergence (JS), whose loss is detailed in Eq 4.7:

$$\mathcal{L}_{\text{KD}} = \frac{1}{2} \mathbb{E}_{\mathbf{Y} \sim p} \left[ \log \frac{p(\mathbf{Y})}{m(\mathbf{Y})} \right] + \frac{1}{2} \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[ \log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')} \right] \tag{4.7}$$

where $p$ is the teacher distribution, $q_\theta$ is the student distribution, $\mathbf{Y}$ and $\mathbf{Y}'$ are sampled from the teacher's and student's distributions and compared with their average $m(\cdot) = \frac{1}{2}p(\cdot) + \frac{1}{2}q_\theta(\cdot)$.

### 4.3.3 Final Learning Objective

The final learning objective the leverages the dataset labels using cross-entropy loss $\mathcal{L}_{\text{CE}}$, but also enforces the use of a specific budget via gate budget loss $\mathcal{L}_{\text{g}}$ and mirrors the behavior of the teacher with the knowledge distillation loss $\mathcal{L}_{\text{KD}}$.Thus, CLSR modules parameters are learned to minimize final loss expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{g}} + \beta \mathcal{L}_{\text{KD}} \tag{4.8}$$

where $\beta$ is a constant defined based on the quality of the teacher, but can also be scheduled or learned (with the add of new constraints for its magnitude).

# 5 Experiments and Results on DistilWhisper

In the former chapter we presented the *DistilWhisper* approach. In this chapter we present how we validate our architecture and the method as a whole, showing that our approach is able to outperform both classical fine-tuning and adapters on `whisper-small`, providing better generalization through light-weight ASR fine-tuning and Knowledge Distillation of the teacher model. Code and models produced in this studied will soon be made available on Hugging Face and Github.

## 5.1 Experimental Setup

In this section we overview our validation setup, that includes choosing the data we use for training and evaluating models, as well as which languages and baselines to consider. We also discuss some code implementation details.

### 5.1.1 Datasets

In order to validate the proposed architecture, we make use of a sample of two widely used massively-multilingual datasets: CommonVoice 13.0 and FLEURS. More details about these datasets are presented on Section 2.5.

In our experiments, we applied downsampling to both the train and validation sets of CV-13, ensuring an equal allocation of training data for each selected language in each experiment. For our primary experiment, we employed 10,000 utterances for training (approximately 14 hours of audio data) and 1,000 for validation. Additionally, we explored variations in dataset size, using downsampled sets of 3,000 and 28,000 utterances in scalability experiments. The selection of data for downsampling was guided by the number of up-votes received by annotators. Notably, we did not apply downsampling to the test set.

For most part of our experiments, FLEURS serves as an invaluable resource for conducting out-of-domain evaluations. It offers a favorable degree of language overlap with the CommonVoice 13.0 dataset (CV-13), making it a suitable choice for comparative analysis. Notably, FLEURS provides an effective out-of-domain setting in the context of ASR evaluation. For instance, while the average number of tokens per sample in CV-13 is 36, FLEURS exhibits a substantially higher average of 97 tokens per sample.

### 5.1.2 Language Selection

In this work we focus on bridging the performance gap for a subset of under-performing languages of the `whisper-small` model through light-weight ASR fine-tuning and Knowledge Distillation of the `whisper-large-v2` model, as proposed in chapter 4. For validating our method, we consider all Whisper languages with a WER gap of more than 11 between large and small models on CV-13.

For our validation experiments we then narrow this list considering: 1) minimum amount of 10k utterances; 2) an overlap with the FLEURS dataset for out-of-domain evaluation. For scalability experiments we loose the first requirement so we can include more diverse set of languages, considering a minimum amount of 3k utterances. We also experiment with the languages in a setting with 28k utterances.

| Resourcefulness | ASR Train data (h) | Languages per setting | | |
|---|---|---|---|---|
| | | 3k | 10k | 28k |
| High-resource | [1000, 5000) | ca, fi, id, pl | ca, pl | ca |
| Mid-to-high-resource | [500, 1000) | uk, vi | uk | - |
| Low-to-mid-resource | [100, 500) | cs, hu, ro, th, ta | cs, hu, th, ta | ta, th |
| Low-resource | [10, 100) | bg, hi, sk, sl | - | - |
| Extremely Low-Resource | (0, 10) | gl | gl | - |

**Table 4** Languages used in the experiments for validation of *DistilWhisper* grouped by resourcefulness.

The final list of languages is: Bulgarian (bg), Catalan (ca), Czech (cs), Finnish (fi), Galician (gl), Hindi (hi), Hungarian (hu), Indonesian (id), Polish (pl), Romanian (ro), Slovak (sk), Slovenian (sl), Tamil (ta), Thai (th), Ukranian (uk), and Vietnamease (vi).[2] These languages belong to 7 distinct language sub-families and exhibit significant variation in terms of their representation within the Whisper training data. This variation extends from a substantial 4,300 hours for certain languages, such as Polish (pl), to a mere 9 hours for languages like Galician (gl). For a detailed overview of these languages and their distribution across the three dataset sizes (3k, 10k, 28k), categorized by their resourcefulness (following the classification proposed on Section 3.1.2), please refer to Table 4. Additionally, Table 5 organizes these languages into groups based on their respective sub-families.

[2] Although Arabic would also qualify considering our criteria, we find that the dialect from FLEURS differs from the ones present on CV-13.

| Sub-families | Languages per setting | | |
|---|---|---|---|
| | 3k | 10k | 28k |
| Slavic (Indo-European) | bg, cs, pl, sk, sl, uk | cs, pl | - |
| Romance (Indo-European) | ca, gl, ro | ca, gl | ca |
| Finno-Ugrian (Uralic) | fi, hu | hu | - |
| Austroasiatic | id, vi | - | - |
| Dravidian | ta | ta | ta |
| Tai (Kra–Dai) | th | th | th |
| Indo-Iranian (Indo-European) | hi | - | - |

**Table 5** Languages used in the experiments for validation of *DistilWhisper* grouped by language sub-families.

### 5.1.3 Models and Baselines

In our evaluation, we assess our approach in comparison to several baseline models. These include the `whisper-small` model, serving as our pre-trained student and starting point, and the `whisper-large-v2` model, acting as the teacher model, and ultimately, as the target goal. Additionally, we explore two fine-tuning (FT) approaches for the student model: standard fine-tuning, where all model weights are updated, and LoRA adaptation, which focuses on refining the feed-forward layer. Moreover, we delve into the effects of the Conditional Language-Specific Routing (CLSR) layer independently, without knowledge distillation (KD), referred to as CLSR-FT. This allows us to isolate the influence of KD from the impact of the CLSR layer on the model's overall robustness.

### 5.1.4 Implementation details

We conducted our experiments using the Transformers library (Wolf et al., 2020) and leveraged the pre-trained weights of both `whisper-small` and `whisper-large-v2` models, sourced from HuggingFace[3] [4]. Unless where stated different, our training protocol consisted of ten epochs, utilizing a learning rate of $10^{-4}$ with linear decay, a one-epoch warm-up phase, a batch size of 16, and a label smoothing factor of 0.1.

For LoRA adaptation, we tested two scenarios: 1) We first adopted the hyperparameters proposed by M. Wang et al. (2023), notably $r = 32$, which is the most commonly

[3] https://huggingface.co/openai/
[4] https://huggingface.co/collections/openai/whisper-release-6501bba2cf999715fd953013

used for this type of adapters; 2) We increase the hidden dimension of the adapters to $r = 64$, so the size of the adapters are comparable to the Language-specific modules on *DistilWhisper*.

In the case of training the CLSR, we set the gate budget ($b$) to 0.5 and the skip-gate probability ($s$) to 0.2. For knowledge distillation (KD), we employed the Jensen–Shannon divergence (JS) with a temperature ($\tau$) of 1, unless when stated in contrary. This was weighted such that the learning objective ($\mathcal{L}$) consisted of the cross-entropy loss ($\mathcal{L}_{\text{CE}}$), the gate loss ($\mathcal{L}_{\text{g}}$), and twice the KD loss ($2\mathcal{L}_{\text{KD}}$): $\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{g}} + 2\mathcal{L}_{\text{KD}}$.

We reported the normalized Word Error Rate (WER) using the Whisper normalization method, with a slight modification to prevent the splitting of numbers and Latin-scripted text into individual characters in languages that do not employ space delimitation (e.g., Thai). Further details, including the modified normalization method, implementation scripts, and model weights, will soon be made available on GitHub and HuggingFace.

Throughout our experiments, we selected the best-performing model based on its WER performance on the downsampled CV-13 validation set.

### 5.2 *DistilWhisper* versus other adaptation approaches

Table 6 presents the results for our first experiment. The top portion presents `whisper-large-v2` (upper bound) and `whisper-small` (lower bound) pre-trained scores, which should not be directly compared to the other adaptation techniques (middle and bottom), as these models were not trained on CV-13 (full out-of-domain setting). The middle portion presents standard fine-tuning (FT) and LoRA adaptation at the feed-forward layers (LoRA-FT). Our results are presented in the bottom: CLSR-FT corresponds to the setting without $\mathcal{L}_{\text{KD}}$, while *DistilWhisper* is the complete setting in which both CLSR and KD losses are leveraged.

For `whisper-small`, we observe that both the standard fine-tuning method (FT) and the LoRA Adapters (LoRA-FT) approaches (middle portion of Table 6) demonstrate the capacity to enhance performance on the in-domain test set (CV-13). However, as anticipated, employing FT leads to a decline in performance on the out-of-domain test set, with an average increase of 1.6. This is likely attributed to catastrophic forgetting, resulting in a tendency to overly specialize in the specific domain. In contrast, LoRA-FT represents a more lightweight adaptation technique that preserves the pre-trained representation. Remarkably, it exhibits improvements in performance on both the in-domain (average decrease of 12.8) and out-of-domain (average decrease of 5.6) test sets when compared to `whisper-small`. Notably, experimenting with a larger hidden

| Common voice 13.0 (in-domain for FT only) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **# params** | **avg** | **ca** | **th** | **ta** | **hu** | **cs** | **pl** | **gl** | **uk** |
| whisper large-v2 | 1.5 B | 14.9 | 16.9 | 9.3 | 17.3 | 18.6 | 14.5 | 8.1 | 19.0 | 15.6 |
| whisper-small | 244 M | 31.4 | 30.1 | 20.3 | 30.1 | 45.5 | 38.6 | 18.8 | 35.7 | 32.3 |
| +FT | 244 M | 22.0 | 19.0 | 10.9 | 17.3 | 30.4 | 29.2 | 21.4 | 19.3 | 28.8 |
| +LoRA-FT (r=32) | 256 M | 18.6 | 15.7 | 9.2 | 15.3 | 30.5 | 25.0 | **15.4** | 12.8 | 24.8 |
| +LoRA-FT (r=64) | 267 M | 18.6 | 15.5 | 9.2 | 15.5 | 30.6 | 25.2 | **15.4** | 13.0 | 24.6 |
| +CLSR-FT | 269 M | 16.4 | 13.9 | 7.4 | 13.6 | 24.9 | 20.9 | 16.0 | **11.2** | 23.5 |
| DistilWhisper | 269 M | **16.1** | **13.8** | **7.2** | **12.5** | **24.1** | **19.9** | 16.1 | 11.6 | **23.2** |
| **FLEURS (out-of-domain)** | | | | | | | | | | |
| **Model** | **# params** | **avg** | **ca** | **th** | **ta** | **hu** | **cs** | **pl** | **gl** | **uk** |
| whisper large-v2 | 1.5 B | 12.6 | 5.6 | 12.6 | 19.3 | 17.9 | 14.4 | 5.9 | 16.8 | 8.3 |
| whisper-small | 244 M | 29.2 | 14.6 | 22.7 | 36.2 | 42.9 | 40.3 | **18.2** | 33.5 | 24.8 |
| +FT | 244 M | 30.8 | 19.1 | 28.2 | 31.6 | 51.3 | 38.9 | 26.1 | 23.2 | 27.9 |
| +LoRA-FT (r=32) | 256 M | 23.6 | 15.5 | 17.6 | 25.5 | 38.5 | 33.4 | 18.5 | 17.7 | 22.3 |
| +LoRA-FT (r=64) | 267 M | 23.6 | 15.7 | 17.6 | 25.7 | 38.2 | 33.9 | 18.5 | 17.3 | **22.1** |
| +CLSR-FT | 269 M | 23.6 | 15.5 | 15.7 | 23.2 | 37.6 | 31.2 | 22.9 | 16.9 | 25.9 |
| DistilWhisper | 269 M | **22.8** | **15.4** | **15.1** | **21.6** | **37.2** | **29.8** | 21.4 | **16.7** | 25.1 |

**Table 6** WER (↓) for the 10k setting with dataset averages (avg) for baselines (top), adaptation approaches (middle), and our method (bottom) for in-domain (CV-13, FT only) and out-of-domain (FLEURS, all) test sets.Best results for `whisper-small` in **bold**.

dimension ($r$) for the LoRA adapters did not yield any perceptible improvement on the average.

Our approach, *DistilWhisper*, yields notable enhancements in performance. When compared to `whisper-small`, it achieves a substantial improvement on in-domain data, with an average decrease of 15.3. This improvement is also evident when compared to LoRA-FT, where an average decrease of 2.2 is observed. Additionally, *DistilWhisper* exhibits superior adaptability in out-of-domain scenarios when contrasted with the original `whisper-small`, resulting in an average increase of 6.4. Furthermore, it demonstrates more effective out-of-domain adaptation capabilities in comparison to LoRA-FT, with an average increase of 0.8. We observe that both versions of our approach, with and without KD, successfully outperform all other adaptation approaches (FT, LoRA-FT) for in-domain and out-of-domain in all languages but two (pl and uk) (bottom portion of Table 6). These findings highlights the robustness of our approach, showcasing that the proposed architecture with the addition of CLSR layers on Whisper provides a strong solution. Notably, all of these improvements are achieved with a mere 25 million parameter overhead during inference (10 % of the original model size).

### 5.3 Impact of knowledge distillation

In this analysis, we compare the two versions of our approach: one entails optimizing a lightweight CLSR-based architecture without Knowledge Distillation (CLSR-FT), while the other incorporates Knowledge Distillation loss (*DistilWhisper*). Across the examined languages, we observe some interesting trends.

Firstly, when considering in-domain performance, as shown in Table 6, the *DistilWhisper* model exhibits a slightly increase in average performance of 0.3 on the WER. The performance is superior in all languages but Polish and Galician. However, when it comes to out-of-domain scenarios, *DistilWhisper* consistently outperforms CLSR-FT across all languages, resulting in an average improvement of 0.8 on the WER. This observation confirms our initial hypothesis that the inclusion of Knowledge Distillation leverages the robustness imparted by the teacher model, preventing over-specialization in the CV-13 domain.

Collectively, these results underscore the effectiveness of our proposed architecture. Notably, we managed to bridge the out-of-domain performance gap between `large-v2` and `small` by a substantial 39%, reducing it from 16.6 to 10.2 (average decrease of 6.5). All of this was achieved with only a modest 10% parameter overhead during inference (25 million parameters).

### 5.4 *DistilWhisper* Scalability

In the previous sections we showed that our architecture improves scores for both in-domain and out-of-domain datasets, compared to other adaptation approaches. In this section we investigate the effectiveness of our method with respect to the amount of data available for training. For this, we select a subset of languages for which we find more training data available on CV-13 (ca, th, ta). Table 7 presents results for our approach in lower-resource training settings (3k utterances; approx. 4 hours), and higher-resource settings (28k utterances; approx. 40 hours). 10k results as well as the results for `whisper-large-v2` and `whisper-small` are repeated from Table 6.

We observe that, as expected, increasing the amount of trainable examples leads to superior ASR performance for both approaches, with the leveraging of KD (*DistilWhisper*) being consistently superior to CLSR-FT and getting closer to close the out-of domain performance gap. For the 28k setup, we are able to reduce the out-of-domain WER gap between `whisper-large-v2` and `whisper-small` by 75.8%, from 12.0 to 2.9.

| | Train size | FLEURS avg | CV-13 avg | FLEURS (out-of-domain) ca | ta | th | CV-13 (in-domain) ca | ta | th |
|---|---|---|---|---|---|---|---|---|---|
| whisper large-v2 | - | 12.5 | 14.5 | 5.6 | 19.3 | 12.6 | 16.9 | 17.3 | 9.3 |
| whisper-small | - | 24.5 | 26.8 | 14.6 | 36.2 | 22.7 | 30.1 | 30.1 | 20.3 |
| +LoRA-FT (r=64) | 3k | 22.7 | 17.0 | 17.7 | 28.6 | 21.7 | 19.4 | 19.0 | 12.5 |
| +CLSR-FT | 3k | 20.4 | 15.2 | 17.8 | **25.4** | 17.9 | 19.2 | 16.7 | 9.7 |
| DistilWhisper | 3k | **20.2** | **14.8** | **17.2** | 25.7 | **17.6** | **18.9** | **15.9** | **9.6** |
| +LoRA-FT (r=64) | 10k | 19.7 | 13.4 | 15.7 | 25.7 | 17.6 | 15.5 | 15.5 | 9.2 |
| +CLSR-FT | 10k | 18.1 | 11.6 | 15.5 | 23.2 | 15.7 | 13.9 | 13.6 | 7.4 |
| DistilWhisper | 10k | **17.4** | **11.2** | **15.4** | **21.6** | **15.1** | **13.8** | **12.5** | **7.2** |
| +LoRA-FT (r=64) | 28k | 17.2 | 11.1 | 13.6 | 23.0 | 15.1 | 12.5 | 13.5 | 7.3 |
| +CLSR-FT | 28k | 15.6 | 9.7 | 13.5 | 19.6 | **13.8** | 11.5 | 11.3 | 6.2 |
| DistilWhisper | 28k | **15.4** | **9.3** | **13.1** | **19.2** | 14.0 | **11.3** | **10.9** | **5.7** |

**Table 7** WER (↓) for different training data sizes (3k, 10k, and 28k utterances) for both in-domain (CV-13) and out-of-domain (FLEURS) test sets. Best results in **bold**.

Furthermore, our approach demonstrates commendable robustness in relation to the quantity of trainable examples. Even with as few as 3,000 utterances (equivalent to 4 hours of training data), we are able to reduce the WER performance gap by 35.8% in out-of-domain data. This suggests that our method holds promise in enhancing ASR performance for low-resource languages, where training data availability is limited.

Across all three settings, our approaches consistently outperform LoRA Adapters by a significant margin. Additionally, it is worth noting that, in nearly all cases within these settings, the inclusion of knowledge distillation proved more beneficial than fine-tuning alone, reinforcing the findings discussed in Section 5.3.

### 5.5 Gate Activation Analysis

To better understand how the model uses routing mechanism, we analyze gate activation statistics on the experiment discussed on Section 5.4 for both CLSR-FT and *DistilWhisper*. This results are presented on Figure 14.

Firstly, we observe a tendency for the models to rely more heavily on the newly introduced Language-Specific modules in out-of-domain scenarios. This could be attributed to the greater complexity and larger sentence sizes prevalent in the FLEURS dataset.

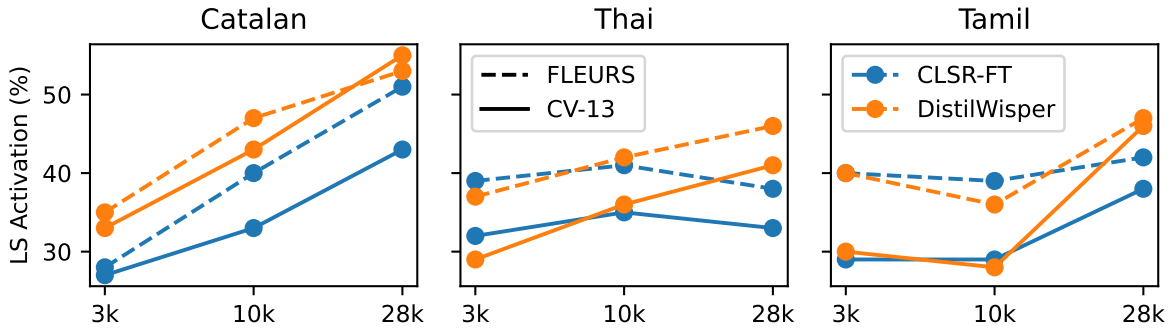


**Figure 14** Ratio of LS layers chosen by the models (CLSR-FT and *DistilWhisper*) depending on (1) amount of training data; (2) in (CV-13) or out-of-domain (FLEURS); (3) language.

Also, as expected, enlarging the training dataset consistently results in more reliable Language-Specific modules, leading to increased utilization of these modules. The only exception for this is Thai at the 28k setup with CLSR-FT, and this might be due to dataset quality and requires further investigation

The comparison of the three languages reveals that Catalan displays a notably higher reliance on Language-Specific routes. This characteristic might be linked to the superior data quality available for Catalan in CV-13, where a substantial number of contributors have contributed to the dataset. Also, the distilled version uses more LS modules, probably because the teacher `whisper-large-v2` is a really good model for this language.

Now for languages with a weaker teacher (Thai, Tamil) we observe that the model may receive contradictory signals at lower-resource settings (3k, 10k), leading to less Language-Specific routing usage with Knowledge Distilation. However, in the higher resource setting (28k), KD usage leads systematically to more reliable Language-Specific module and therefore higher LS routing.

Finally, we observe a common trend across the three languages - models tend to employ more Language-Specific routes when learning with Knowledge Distillation (*DistilWhisper* vs. CLSR-FT). This suggests that KD imparts valuable information and enhances the out-of-domain generalization capabilities of the learned Language-Specific representation.

| Common voice 13.0 (in-domain for FT only) | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **avg** | **bg** | **ca** | **cs** | **fi** | **gl** | **hi** | **hu** | **id** | **pl** | **ro** | **sk** | **sl** | **ta** | **th** | **uk** |
| whisper large-v2 | 17.0 | 19.9 | 16.9 | 14.5 | 14.4 | 19.0 | 24.6 | 18.6 | 8.5 | 8.1 | 15.8 | 31.9 | 20.6 | 17.3 | 9.3 | 15.6 |
| whisper-small | 34.2 | 44.8 | 30.1 | 38.6 | 30.5 | 35.7 | 43.6 | 45.5 | 22.5 | 18.8 | 33.2 | 42.0 | 45.5 | 30.1 | 20.3 | 32.3 |
| +CLSR-FT | 22.9 | 26.1 | 19.2 | **25.7** | 25.1 | 15.3 | 18.8 | 31.6 | 19.2 | **18.3** | 23.4 | **36.6** | 28.6 | 16.7 | 9.7 | **29.5** |
| DistilWhisper | **22.6** | **25.9** | **18.9** | 26.2 | **24.8** | **14.7** | **18.3** | **31.0** | **18.6** | 18.6 | **21.5** | 36.8 | **27.7** | **15.9** | **9.6** | 30.0 |
| **FLEURS (out-of-domain)** | | | | | | | | | | | | | | | | |
| **Model** | **avg** | **bg** | **ca** | **cs** | **fi** | **gl** | **hi** | **hu** | **id** | **pl** | **ro** | **sk** | **sl** | **ta** | **th** | **uk** |
| whisper large-v2 | 13.7 | 14.6 | 5.6 | 14.4 | 9.7 | 16.8 | 23.8 | 17.9 | 7.1 | 5.9 | 14.4 | 11.7 | 23.1 | 19.3 | 12.6 | 8.3 |
| whisper-small | 32.8 | 39.9 | 14.6 | 40.3 | 26.8 | 33.5 | 47.9 | 42.9 | **18.6** | **18.2** | 34.6 | 35.8 | 54.5 | 36.2 | 22.7 | 24.8 |
| +CLSR-FT | **29.2** | 43.8 | 17.8 | 35.4 | 33.7 | 19.8 | 22.8 | **40.1** | 19.0 | 21.9 | 33.4 | 35.3 | 50.8 | **25.4** | 17.9 | **21.6** |
| DistilWhisper | **29.2** | **42.8** | **17.2** | **35.6** | **32.0** | **18.7** | **21.8** | 41.1 | 19.1 | 21.9 | **33.1** | **35.2** | **50.5** | 25.7 | **17.6** | 25.9 |

| | FLEURS (out-of-domain) | | | | | CV-13 (in-domain) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **High** | **Mid-to-high** | **Low-to-mid** | **Low** | **Extremely Low** | **High** | **Mid-to-high** | **Low-to-mid** | **Low** | **Extremely Low** |
| whisper large-v2 | 7.1 | 8.3 | 15.7 | 18.3 | 16.8 | 12.0 | 15.6 | 15.1 | 24.3 | 19.0 |
| whisper-small | **19.6** | 24.8 | 35.3 | 44.5 | 33.5 | 25.5 | 32.3 | 33.5 | 44.0 | 35.7 |
| +CLSR-FT | 23.1 | **21.6** | **30.4** | 38.2 | 19.8 | 20.4 | **29.5** | 21.4 | 27.5 | 15.3 |
| DistilWhisper | 22.5 | 25.9 | **30.6** | **37.6** | **18.7** | **20.2** | 30.0 | **20.8** | **27.2** | **14.7** |

**Table 8** WER (↓) for the 3k setting with dataset averages (avg) for baselines (top), and our method (bottom) for in-domain (CV-13, FT only - higher portion) and out-of-domain (FLEURS, all - middle portion) test sets. On the lower portion, the same results are grouped by resourcefulness. Best results for `whisper-small` in **bold**.

### 5.6 Considerations on the Resourcefulness

Our observations so far indicate that both versions of our approach, with and without knowledge distillation (KD), demonstrate consistent outperformance over all other adaptation methods (FT and LoRA-FT). This improvement holds true for both in-domain and out-of-domain scenarios across all languages, with only two exceptions on the 10k setting (Polish and Ukrainian), as indicated in the lower portion of Table 6. The challenges encountered in these two languages can be attributed to their higher resource status, with Polish being a high-resource language and Ukrainian categorized as mid-to-high resource, as detailed in Table 4.

In order to deepen this analysis, we conducted experiments across a broader range of languages, widening to those with a minimum of 3,000 utterances available for training. The outcomes of these experiments are presented in Table 8, where we have also aggregated the results into resourcefulness clusters (in the lower portion) based on the classification provided in Table 4.

Examining the results, we observed that more substantial out-of-domain improvements are seem in languages with lower resource availability (Low-to-mid, Low and Extremely low-resource clusters). This aligns with the initial motivation behind our work, which aimed to address the *curse of multilinguality*. We expect that lower resource languages experience a more significant impact from this phenomenon during the pre-training of `whisper-small`. Consequently, they significantly benefit more from the integration of language-specific modules in the feature domain.

In contrast, for languages with higher resource availability, further enhancements may be necessary, such as adjustments to attention weights (corresponding to the time domain). This is due to the fact that the original model already performs reasonably well. Additionally, achieving better out-of-domain performance may require a larger training dataset. This is exemplified by the case of Catalan presented in Table 7. In this case, CLSR modules yielded superior performance than original `whisper-small` only in the case trained with 28,000 utterances, losing to its starting point for 3,000 and 10,000 training utterances.

### 5.7 Effect of temperature and distillation loss

In this set of experiments, our goal is to examine the impact of the chosen distillation optimization on the results. We start by exploring the effect of temperature. Temperature plays a crucial role in determining the learning behavior of the model. A lower temperature, like 1, tends to make the learning focus primarily on replicating the first

option from the teacher's logits for each token. Conversely, a higher temperature, such as 3 or 4, encourages the learning to take into account the other options, thereby mitigating the cost from incorrect predictions. However, this approach may lead to over-smoothing of the distribution and a reduced ability to effectively rank similar logits.

| **Common voice 13.0 (in-domain)** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **avg** | **ca** | **th** | **ta** | **hu** | **cs** | **pl** | **gl** | **uk** |
| JS w/ $\tau = 1$ | **16.1** | **13.8** | **7.2** | **12.5** | **24.1** | **19.9** | **16.1** | **11.6** | **23.2** |
| JS w/ $\tau = 3$ | 16.3 | 14.1 | 7.5 | 13.1 | 23.5 | 21.1 | 16.2 | **11.6** | 23.6 |
| **FLEURS (out-of-domain)** | | | | | | | | | |
| | **avg** | **ca** | **th** | **ta** | **hu** | **cs** | **pl** | **gl** | **uk** |
| JS w/ $\tau = 1$ | **22.8** | **15.4** | **15.1** | 21.6 | 37.2 | **29.8** | **21.4** | **16.7** | **25.1** |
| JS w/ $\tau = 3$ | 23.4 | 17.0 | 15.6 | **21.5** | **36.0** | 31.4 | 22.4 | 16.8 | 26.2 |

**Table 9** WER (↓) for the 10k setting with dataset averages (avg) for JS loss with temperatures 1 and 3, for in-domain (CV-13, FT only - higher portion) and out-of-domain (FLEURS, all - lower portion) test sets. Best results in **bold**.

Tables 9 and 10 present the results of comparing different temperatures (1 or 3) with the Jensen–Shannon loss for both the 10k and 28k settings. These results reveal that using a temperature of 1 generally results in improved in-domain and out-of-domain performance compared to a temperature of 3. However, for Tamil and Hungarian, temperature 3 showed better out-of-domain performance. These results suggest that `whisper-large-v2` serves as an effective teacher, justifying the use of a temperature of 1. Nevertheless, the optimal temperature value may vary depending on the quality of the teacher model for each specific language.

| | **FLEURS** | **CV-13** | **FLEURS (out-of-domain)** | | | **CV-13 (in-domain)** | | |
|---|---|---|---|---|---|---|---|---|
| | **avg** | **avg** | **ca** | **ta** | **th** | **ca** | **ta** | **th** |
| JS w/ $\tau = 1$ | **15.4** | **9.3** | **13.1** | 19.2 | 14.0 | **11.3** | **10.9** | **5.7** |
| JS w/ $\tau = 3$ | 16.3 | 9.7 | 14.8 | 20.1 | 14.1 | 11.8 | 11.3 | 5.9 |
| KL w/ $\tau = 1$ | 15.6 | 10.8 | 14.6 | **18.7** | **13.3** | 14.9 | 11.3 | 6.2 |
| KL w/ $\tau = 3$ | 16.5 | 9.7 | 15.8 | 19.8 | 14.0 | 12.2 | 11.1 | 5.9 |

**Table 10** WER (↓) for different training data sizes (3k, 10k, and 28k utterances) for JS and KL losses for temperatures 1 and 3 for both in-domain (CV-13) and out-of-domain (FLEURS) test sets. Best results in **bold**.

Table 10 also compares the use of the Jensen–Shannon (JS) loss with the traditional Kullback–Leibler (KL) loss discussed in Section 2.4, specifically for the 28k setting. Once again, the results favor a temperature of 1 in both cases, with a slight advantage for the JS loss against KL, primarily driven by Catalan out-of-domain performance. This advantage is more pronounced in in-domain performance. These findings indicate the presence of the *mode-averaging problem* introduced in Section 2.4, although they are not definitive. They raise questions about whether these behaviors change when working with larger or smaller fine-tuning datasets and different levels of language resourcefulness. Unfortunately, due to time constraints, we could not explore these aspects in this study, leaving them as potential directions for future research.

### 5.8 Multi-domain training

In our final experiment, we delve into the impact of incorporating the train split of FLEURS dataset into our training data in the previously explored settings. The objective here is to use the validated architecture to generate models that would be more beneficial to the scientific community. In real-world scenarios, the models developed here are likely to be utilized in domains other than FLEURS or CV-13, so the hypothesis is that training on more than one dataset yields a better model.

| | | | **Common voice 13.0** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Train data** | **avg** | **ca** | **th** | **ta** | **hu** | **cs** | **pl** | **gl** | **uk** |
| whisper large-v2 | - | 14.9 | 16.9 | 9.3 | 17.3 | 18.6 | 14.5 | 8.1 | 19.0 | 15.6 |
| whisper-small | - | 31.4 | 30.1 | 20.3 | 30.1 | 45.5 | 38.6 | 18.8 | 35.7 | 32.3 |
| DistilWhisper | CV10k | 16.1 | 13.8 | 7.2 | 12.5 | 24.1 | 19.9 | 16.1 | 11.6 | 23.2 |
| +CLSR-FT | CV10k + F | 15.5 | 15.1 | 6.8 | 12.4 | 21.9 | 18.4 | 16.3 | 11.3 | 22.2 |
| DistilWhisper | CV10k + F | **14.6** | **13.2** | **6.4** | **11.6** | **21.6** | **15.3** | **15.8** | **11.2** | **21.6** |
| | | | **FLEURS** | | | | | | | |
| **Model** | **Train data** | **avg** | **ca** | **th** | **ta** | **hu** | **cs** | **pl** | **gl** | **uk** |
| whisper large-v2 | - | 12.6 | 5.6 | 12.6 | 19.3 | 17.9 | 14.4 | 5.9 | 16.8 | 8.3 |
| whisper-small | - | 29.2 | 14.6 | 22.7 | 36.2 | 42.9 | 40.3 | 18.2 | 33.5 | 24.8 |
| DistilWhisper | CV10k | 22.8 | 15.4 | 15.1 | 21.6 | 37.2 | 29.8 | 21.4 | 16.7 | 25.1 |
| +CLSR-FT | CV10k + F | 17.2 | **11.8** | 10.1 | 16.0 | 28.1 | 23.2 | **17.1** | 12.9 | 18.7 |
| DistilWhisper | CV10k + F | **16.7** | 11.9 | **9.4** | **14.6** | **27.7** | **22.1** | 17.7 | **12.7** | **17.3** |

**Table 11** WER (↓) for the setting trained with 10k from CV-13 and FLEURS with dataset averages (avg) for baselines (top), adaptation approaches (middle), and our method (bottom) CV-13 and FLEURS test sets (both in-domain). Best results for `whisper-small` in **bold**.

Table 11 showcases the outcomes of training the model in a setting involving 10k sentences from CV-13 along with the entire FLEURS train split. In this setting, we once

again experiment with CLSR fine-tuning. For reference, the table also presents results from section 5.2. The results reaffirm better performance for the setting with Knowledge Distillation compared to CLSR-FT. More significantly, the results demonstrate a substantial improvement within the domain when FLEURS is incorporated as part of the training dataset. Training with FLEURS reduces the WER on CV-13 by 1.5. This improvement is likely due to FLEURS' greater sentence complexity and larger average token count per line, contributing to enhanced training data diversity.

In table 12, we repeat the same experiment using settings with 3k and 28k sentences from CV-13, both added to the full FLEURS dataset. The results allow us to draw the same conclusions: the addition of out-of-domain training data (FLEURS) results in superior in-domain generalization on CV-13. Nevertheless, it is evident that the size of the training data remains a limiting factor, as CV3k+F (approximately 6k sentences) was insufficient to surpass CV10k alone, and similarly for CV10k+F (around 13k sentences) in comparison to CV28k alone.

In this section, we have presented the best models attainable for each setting using these two datasets. These models will be made open-source, and we hope they contribute to the development of speech recognition applications in these languages.

| | | | Common voice 13.0 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Train data** | **avg** | **bg** | **ca** | **cs** | **fi** | **gl** | **hi** | **hu** | **id** | **pl** | **ro** | **sk** | **sl** | **ta** | **th** | **uk** |
| whisper large-v2 | - | 17.0 | 19.9 | 16.9 | 14.5 | 14.4 | 19.0 | 24.6 | 18.6 | 8.5 | 8.1 | 15.8 | 31.9 | 20.6 | 17.3 | 9.3 | 15.6 |
| whisper-small | - | 34.2 | 44.8 | 30.1 | 38.6 | 30.5 | 35.7 | 43.6 | 45.5 | 22.5 | 18.8 | 33.2 | 42.0 | 45.5 | 30.1 | 20.3 | 32.3 |
| DistilWhisper | CV3k | 22.6 | 25.9 | 18.9 | 26.2 | 24.8 | 14.7 | 18.3 | 31.0 | 18.6 | 18.6 | 21.5 | 36.8 | 27.7 | 15.9 | 9.6 | 30.0 |
| DistilWhisper | CV3k + F | **19.3** | **21.8** | **15.0** | **21.7** | **22.4** | **14.2** | **15.8** | **26.4** | **17.0** | **17.2** | **18.0** | **29.3** | **22.9** | **13.4** | **7.8** | **27.0** |
| | | | **FLEURS** | | | | | | | | | | | | | | |
| | **Train data** | **avg** | **bg** | **ca** | **cs** | **fi** | **gl** | **hi** | **hu** | **id** | **pl** | **ro** | **sk** | **sl** | **ta** | **th** | **uk** |
| whisper large-v2 | - | 13.7 | 14.6 | 5.6 | 14.4 | 9.7 | 16.8 | 23.8 | 17.9 | 7.1 | 5.9 | 14.4 | 11.7 | 23.1 | 19.3 | 12.6 | 8.3 |
| whisper-small | - | 32.8 | 39.9 | 14.6 | 40.3 | 26.8 | 33.5 | 47.9 | 42.9 | 18.6 | 18.2 | 34.6 | 35.8 | 54.5 | 36.2 | 22.7 | 24.8 |
| DistilWhisper | CV3k | 29.2 | 42.8 | 17.2 | 35.6 | 32.0 | 18.7 | 21.8 | 41.1 | 19.1 | 21.9 | 33.1 | 35.2 | 50.5 | 25.7 | 17.6 | 25.9 |
| DistilWhisper | CV3k + F | **18.3** | **21.0** | **12.3** | **24.2** | **19.7** | **13.9** | **13.5** | **29.0** | **13.0** | **16.6** | **21.4** | **19.4** | **27.5** | **15.1** | **10.3** | **18.1** |

| | **Training** | **FLEURS** | **CV-13** | **FLEURS** | | | **CV-13** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Data** | **avg** | **avg** | **ca** | **ta** | **th** | **ca** | **ta** | **th** |
| whisper large-v2 | - | 12.5 | 14.5 | 5.6 | 19.3 | 12.6 | 16.9 | 17.3 | 9.3 |
| whisper-small | - | 24.5 | 26.8 | 14.6 | 36.2 | 22.7 | 30.1 | 30.1 | 20.3 |
| DistilWhisper | CV28k | 15.4 | 9.3 | 13.1 | 19.2 | 14.0 | 11.3 | 10.9 | 5.7 |
| DistilWhisper | CV28k + F | **11.4** | **9.0** | **10.8** | **14.2** | **9.4** | **10.9** | **10.5** | **5.6** |

**Table 12** WER (↓) for the 3k setting with dataset averages (avg) for baselines (top), and our method (bottom) for in-domain (CV-13, FT only - higher portion) and out-of-domain (FLEURS, all - middle portion) test sets. On the lower portion, the same results are grouped by resourcefulness. Best results for `whisper-small` in **bold**.

# 6 Conclusion

This internship focused on investigating bias on Whisper, a family of large speech models, specifically examining speaker-related (gender, age, accent) and model-related (model size, resourcefulness, similar languages) biases. Additionally, we explored whether these biases are mitigated or exacerbated by quantization and proposed an alternative compression approach.

Our findings revealed that Whisper exhibits both speaker-related and model-related biases. Speaker-related biases are kept unchanged after quantization, while model-related biases are amplified by this compression technique. Low-resource languages are particularly more affected, and smaller models experience significant performance degradation. This is concerning because current parameter-efficient approaches typically apply quantization uniformly across models, introducing unintended bias.

To address this challenge, we introduced *DistilWhisper*, a parameter-efficient distillation approach that enhances the performance of `whisper-small` by transferring the robustness of `whisper-large-v2` into a smaller model. This is achieved by incorporating language-specific gated modules and jointly optimizing ASR fine-tuning and knowledge distillation losses. Our results consistently showed performance improvements across various languages and test sets, with minimal parameter increase during inference. We believe this approach will democratize the use of Whisper models, making them accessible to a wider audience of researchers and practitioners. This approach was organized as a paper submitted to the conference ICASSP 2024 (Ferraz et al., 2024). Code and models produced in this study will be made available soon on Hugging Face and Github.

## 6.1 Future Work

There are several promising directions for future research in this area. Firstly, it would be beneficial to expand upon the analysis presented in Chapter 3, including an investigation into other quantization methods, such as 4-bit quantization. Exploring these methods across various model families would help determine if the conclusions drawn here are applicable more broadly. This could present an important contribution to the community and ensure the correct usage of these techniques.

Additionally, further research into the *DistilWhisper* approach could yield valuable insights. Examining the effects of several hyperparameters, such as gate budget, KD loss weight, and temperature, would provide a deeper understanding of the approach's

behavior. This exploration could help find the best setting for optimal performance of the approach.

Furthermore, it would be valuable to assess the impact of the proposed approach in multitasking beyond transcription (ASR), particularly in speech translation. Investigating whether language-specific paths can enhance translation performance to English, and exploring the potential for achieving new zero-shot capabilities in many-to-many translation scenarios, could open up exciting possibilities for the field.

## References

Antonios, A., Loc, B., Bentivogli, L., Boito, M. Z., Ondřej, B., Cattoni, R., Anna, C., Georgiana, D., Kevin, D., Maha, E., et al. (2022). Findings of the iwslt 2022 evaluation campaign. *Proc. of the 19th Int. Conf. on Spoken Language Translation (IWSLT 2022)*.

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., & Weber, G. (2020). Common voice: A massively-multilingual speech corpus. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 4218–4222. https://aclanthology.org/2020.lrec-1.520

Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G., Cherry, C., Macherey, W., Chen, Z., & Wu, Y. (2019). Massively multilingual neural machine translation in the wild: Findings and challenges.

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., & Auli, M. (2022). XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. *Proc. Interspeech 2022*, 2278–2282. https://doi.org/10.21437/Interspeech.2022-143

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). Wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, *33*, 12449–12460.

Boito, M. Z., Besacier, L., Tomashenko, N., & Estève, Y. (2022). A study of gender impact in self-supervised models for speech-to-text systems. *arXiv preprint arXiv:2204.01397*.

Bondarenko, Y., Nagel, M., & Blankevoort, T. (2021). Understanding and overcoming the challenges of efficient transformer quantization. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7947–7969. https://doi.org/10.18653/v1/2021.emnlp-main.627

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*.

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2021). Unsupervised Cross-Lingual Representation Learning for Speech Recognition. *Proc. Interspeech 2021*, 2426–2430. https://doi.org/10.21437/Interspeech.2021-329

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451.

Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., & Bapna, A. (2023). Fleurs: Few-shot learning evaluation of universal representations of speech. *2022 IEEE SLT*, 798–805.

Costa-jussà, M. R., Basta, C., & Gállego, G. I. (2022). Evaluating gender bias in speech translation. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2141–2147.

Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, *35*, 30318–30332.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). Qlora: Efficient fine-tuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Dettmers, T., & Zettlemoyer, L. (2023). The case for 4-bit precision: K-bit inference scaling laws. *International Conference on Machine Learning*, 7750–7774.

Evain, S., Nguyen, M. H., Le, H., Boito, M. Z., Mdhaffar, S., Alisamir, S., Tong, Z., Tomashenko, N., Dinarelli, M., Parcollet, T., et al. (2021). Task agnostic and task specific self-supervised learning from speech with lebenchmark. *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*.

Feng, T., & Narayanan, S. (2023). Peft-ser: On the use of parameter efficient transfer learning approaches for speech emotion recognition using pre-trained speech models. *arXiv preprint arXiv:2306.05350*.

Ferraz, T. P., Zanon Boito, M., Brun, C., & Nikoulina, V. (2024). Multilingual Distilwhisper: Efficient Distillation of Multi-Task Speech Models Via Language-Specific Experts. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 10716–10720. https://doi.org/10.1109/ICASSP48485.2024.10447520

Gandhi, S., Von Platen, P., & Rush, A. M. (2022). Esb: A benchmark for multi-domain end-to-end speech recognition. *arXiv preprint arXiv:2210.13352*.

Go, D., Korbak, T., Kruszewski, G., Rozen, J., Ryu, N., & Dymetman, M. (2023). Aligning language models with preferences through $f$-divergence minimization. *Proceedings of the 40th International Conference on Machine Learning*. https://proceedings.mlr.press/v202/go23a.html

Gow-Smith, E., Berard, A., Zanon Boito, M., & Calapodescu, I. (2023). NAVER LABS Europe's multilingual speech translation systems for the IWSLT 2023 low-resource track. *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, 144–158. https://doi.org/10.18653/v1/2023.iwslt-1.10

Goyal, N., Du, J., Ott, M., Anantharaman, G., & Conneau, A. (2021). Larger-scale transformers for multilingual masked language modeling.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., & Fan, A. (2022). The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, *10*, 522–538.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*. http://arxiv.org/abs/1503.02531

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-efficient transfer learning for nlp. *International Conference on Machine Learning*, 2790–2799.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3451–3460.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. *International Conference on Learning Representations*. https://openreview.net/forum?id=nZeVKeeFYf9

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, *18*(1), 6869–6898.

Jain, R., Barcovschi, A., Yiwere, M., Corcoran, P., & Cucu, H. (2023). Adaptation of whisper models to child speech recognition. *arXiv preprint arXiv:2307.13008*.

Laskar, M. T. R., Hoque, E., & Huang, J. X. (2022). Domain Adaptation with Pre-trained Transformers for Query-Focused Abstractive Text Summarization. *Computational Linguistics*, *48*(2), 279–320. https://doi.org/10.1162/coli_a_00434

Le, H., Pino, J., Wang, C., Gu, J., Schwab, D., & Besacier, L. (2021). Lightweight adapter tuning for multilingual speech translation. *Proc. of the 59th Annual Meeting of the ACL and the 11th Int. Joint Conf. on Natural Language Processing*.

Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, *461*, 370–403.

Menghani, G. (2023). Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, *55*(12), 1–37.

Mohammadshahi, A., Nikoulina, V., Berard, A., Brun, C., Henderson, J., & Besacier, L. (2022). SMaLL-100: Introducing shallow multilingual machine translation model for low-resource languages. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2022.emnlp-main.571

Pfeiffer, J., Goyal, N., Lin, X., Li, X., Cross, J., Riedel, S., & Artetxe, M. (2022). Lifting the curse of multilinguality by pre-training modular transformers. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3479–3495. https://doi.org/10.18653/v1/2022.naacl-main.255

Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., Elkahky, A., Ni, Z., Vyas, A., Fazel-Zarandi, M., et al. (2023). Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. *International Conference on Machine Learning*, 28492–28518.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Radhakrishnan, S., Yang, C.-H. H., Khan, S. A., Kiani, N. A., Gomez-Cabrero, D., & Tegner, J. N. (2023). A parameter-efficient learning approach to arabic dialect identification with pre-trained general-purpose speech model. *arXiv preprint arXiv:2305.11244*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter.

Shen, Z., Guo, W., & Gu, B. (2023). Language-universal adapter learning with knowledge distillation for end-to-end multilingual speech recognition. *arXiv preprint arXiv:2303.01249*.

Shi, J., Berrebbi, D., Chen, W., Chung, H.-L., Hu, E.-P., Huang, W. P., Chang, X., Li, S.-W., Mohamed, A., Lee, H.-y., et al. (2023). Ml-superb: Multilingual speech universal performance benchmark. *arXiv preprint arXiv:2305.10615*.

Thomas, B., Kessler, S., & Karout, S. (2022). Efficient adapter transfer of self-supervised speech models for automatic speech recognition. *ICASSP 2022-2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Tomanek, K., Zayats, V., Padfield, D., Vaillancourt, K., & Biadsy, F. (2021). Residual adapters for parameter-efficient asr adaptation to atypical and accented speech. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6751–6760.

Valk, J., & Alumäe, T. (2021). VoxLingua107: a dataset for spoken language recognition. *2021 IEEE Spoken Language Technology Workshop (SLT)*, 652–658.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, M., Li, Y., Guo, J., Qiao, X., Li, Z., Shang, H., Wei, D., Tao, S., Zhang, M., & Yang, H. (2023). WhiSLU: End-to-End Spoken Language Understanding with Whisper. *Proc. INTERSPEECH 2023*, 770–774. https://doi.org/10.21437/Interspeech.2023-1505

Wang, Y., Li, J., Wang, H., Qian, Y., Wang, C., & Wu, Y. (2022). Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7097–7101.

Wen, Y., Li, Z., Du, W., & Mou, L. (2023). F-divergence minimization for sequence-level knowledge distillation. *ACL*. https://doi.org/10.18653/v1/2023.acl-long.605

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Wu, H., Judd, P., Zhang, X., Isaev, M., & Micikevicius, P. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*.

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., Lee, K.-t., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H.-y. (2021). SUPERB: Speech Processing Universal PERformance Benchmark. *Proc. Interspeech 2021*, 1194–1198. https://doi.org/10.21437/Interspeech.2021-1775

Zhang, B., Bapna, A., Sennrich, R., & Firat, O. (2021). Share or not? learning to schedule language-specific capacity for multilingual translation. *International Conference on Learning Representations*. https://openreview.net/forum?id=Wj4ODo0uyCF

Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., Chen, Z., Chen, N., Li, B., Axelrod, V., Wang, G., et al. (2023). Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.

# Appendices

## A Paper submitted to ICASSP 2024: "Multilingual DistilWhisper: Efficient Distillation of Multi-task Speech Models via Language-Specific Experts"

Paper submitted to ICASSP 2024.