university of groningen

campus fryslân

Parameter-Efficient Fine-Tuning on Multilingual ASR Whisper Model for Frisian

Xueying Liu

University of Groningen - Campus Fryslân

Parameter-Efficient Fine-Tuning on Multilingual ASR Whisper Model for Frisian

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Voice Technology
at University of Groningen under the supervision of
Dr. Shekhar Nayak (Voice Technology, University of Groningen)

Xueying Liu (S5521904)

11 June, 2024

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr. Shekhar Nayak, for his invaluable guidance on this thesis topic and his support throughout the research. I also thank Phat Do for his assistance with technical problems and Dr. Matt Coler for his feedback on the thesis proposal.

I am grateful to my colleagues in the Voice Technology programme for their advice. Special thanks to Yanpei Ouyang for closely collaborating on our theses and for cheering me up with her positive spirit.

Lastly, I am deeply thankful to my family, especially my husband, for his unwavering support and deep trust in me.

Abstract

Despite the proven multi-language competencies of Whisper, the model faces challenges when recognizing low-resource languages (LRLs). The typical way to improve its performance on LRLs is to fully fine-tune the model with the additional target LRLs data. Still, due to the extensive parameter sets of the model and a limited amount of data, this approach is resource-intensive and prone to overfit. To compensate for the tremendous computational cost of the full fine-tuning and overfitting problem, parameter-efficient fine-tuning (PEFT) such as Low-Rank Adaptation (LoRA), is proposed as a feasible solution. In this work, I examined the effectiveness of LoRA on the Whisper model for low-resource language Frisian. The result showed that with only 1.4% of model parameters and less GPU memory, LoRA achieved comparable word error rate (WER) performance to full fine-tuning in Frisian. I also found that low-resource languages benefited more from LoRA than high-resource languages. This study brings valuable insights for practical ASR system development toward efficiency and inclusion, particularly in multilingual and low-resource contexts.

Keywords: Parameter-Efficient Fine-Tuning, LoRA, Low-Resource Languages, Frisian, Multilingual ASR

Content

1 Introduction

Automatic Speech Recognition (ASR) systems, pivotal in applications from virtual assistants to captioning services, hinge on advanced large pre-trained multilingual speech models like XLS-R (Babu et al., 2021), Whisper (Radford et al., 2022a), USM (Zhang et al., 2023) and MMS (Pratap et al., 2023), characterized by their extensive parameter sets. Taking Whisper as an example, the smallest Whisper model has 39 million parameters, while the largest model which also gives the best performance, has as large as 1550 million parameters. Despite the proven multi-language and multi-task competencies, these multilingual models face challenges in recognizing low-resource languages (LRLs). The Whisper $large$ model can recognize high-resource languages such as English, German, and Spanish with less than 5% word error rate (WER), whereas it performs very poorly on many LRLs with up to 160% WER.

To improve the performance of LRLs, pre-trained multilingual ASR models can be fully fine-tuned with speech data from the target LRLs. Thanks to the enormous amount of shared knowledge across multiple languages in multilingual ASR models, the full fine-tuning strategy proves to be very effective in enabling pre-trained multilingual models to outperform the monolingual baselines for LRLs (Pratap, Sriram, et al., 2020). However, this approach poses unique challenges in training and deploying ASR systems. Fully fine-tuning a model is very computationally expensive because of the extensive parameterization of the model. After full fine-tuning, storing and deploying the downstream fine-tuned model also takes a large space, because the downstream model will have almost the same size as the large pre-trained model. Moreover, using small size of training datasets to update all model parameters, which is often the case for low-resource languages, is prone to be overfitting (Xu et al., 2023).

To address these huge numbers of parameters and potential overfitting challenges, the parameter-efficient fine-tuning (PEFT) is proposed as a feasible solution to compensate for the tremendous computational cost of full fine-tuning. This method is first introduced in the field of computer vision (Rebuffi et al., 2017) and then is successfully applied to the field of natural language processing (NLP) (Houlsby et al., 2019) and machine translation (Le et al., 2021). There are a variety of PEFT methods such as LoRA (Hu et al., 2021), Residual Adaptor (Rebuffi et al., 2017), Prefix Tuning (X. L. Li & Liang, 2021), Prompt Tuning (Lester et al., 2021) and so on, but the main idea behind them stays the same, that is to reduce the number of trainable parameters while maintaining the comparable performance to the full fine-tuning. During the training process, the backbone of the pre-trained model is frozen and only a small number of parameters are updated. The advantage of this process is twofold. First, the pre-trained model can be adapted to downstream tasks with only a small number of parameters, which greatly reduces training costs. Second, the representations in the large pre-trained model can be preserved, enhancing the robustness of downstream models and avoiding the risk of catastrophic forgetting (Xu et al., 2023).

Recently, one of the PEFT methods, Low Rank Adaptation (LoRA) has received increased attention (Hu et al., 2021). It is first introduced in the area of NLP and it performs on par or even better than full fine-tuning at. By freezing the pre-trained model weights and injecting trainable rank decomposition matrices into each layer of the transformer architecture, LoRA greatly reduces the number of trainable parameters for downstream tasks like other PEFT methods do. In addition to the one common advantage of PEFT methods that reduces the number of trainable parameters, the key advantage of LoRA also lies in the fact that it is highly scalable, allowing switching among many LoRA modules for different tasks while sharing a base pre-trained model. Another advantage is that it does not introduce inference latency.

These features of LoRA show great potential in the case of the multilingual ASR model for low-resource language recognition. Multiple low-resource languages can be trained with multiple LoRA modules for swiftly switching among languages, while the model still benefits from the powerful shared knowledge of the pre-trained model. Meanwhile, the entire training process brings much less computational burden and does not slow down the inference stage. I am therefore motivated to investigate PEFT strategies, specifically LoRA, as a method to circumvent limitations while adapting the multilingual ASR Whisper model to low-resource languages.

1.1 Research Questions and Hypotheses

Despite the efficiency of PEFT in various contexts, its application to multilingual ASR models, especially for LRLs such as Frisian, remains less explored. This gap is notable given the increasing need for efficient, scalable ASR systems capable of supporting Frisian and a broad spectrum of languages. I aim to fill this gap by investigating the effectiveness of LoRA on the Whisper model for low-resource language Frisian. More specifically, I will compare the performance between LoRA and full fine-tuning in adapting the Whisper model to perform the Frisian ASR task. To this end, the research question is formulated as follows:

> *How does tuning the Whisper model with LoRA affect the WER, number of trainable parameters, and GPU memory usage for Frisian compared to fully fine-tuning the Whisper model?*

In light of the previous discussion, I anticipate that tuning the Whisper model for Frisian with LoRA will result in a significant reduction in WER, which will be the comparable performance to full fine-tuning. Meanwhile, LoRA will utilize less trainable parameters and less GPU memory than full fine-tuning.

This work contributes to the field of ASR in several ways. First, it is the very first study examining and adapting the Whisper model for Frisian, testifying to the cross-language learning ability of Whisper. Second, it deepens the theoretical understanding of PEFT's applicability in ASR models by presenting a comprehensive review of PEFT methods from the perspective of speech related research. Last, it conducts extensive experiments to investigate the effectiveness of one PEFT method, namely LoRA, specifically examining its impact on parameter efficiency, WER, and GPU memory for a pre-trained Whisper model. Such evaluation would offer valuable insights for practical ASR system development and deployment, particularly in multilingual and low-resource contexts.

1.2 Thesis Structure

The structure of this thesis will be as follows. I will provide the literature review regarding large pre-trained multilingual ASR models, full fine-tuning and PEFT methods and Frisian ASR in Chapter 2. The methodology about datasets, models, evaluation metrics, experiments set-ups and ethical issues will be presented in Chapter 3. In Chapter 4, I will showcase the results and discuss them in detail. Chapter 5 will conclude this study while pointing out limitations and future directions.

2 Literature Review

In this chapter, I will briefly introduce what are the large pre-trained Multilingual ASR models and the typical technique of adapting them for downstream tasks, i.e. full fine-tuning method. Then I will give an overview of PEFT methods and their usage in the speech domain. This chapter will end by introducing the state-of-the-art Frisian ASR models.

2.1 Large Pre-Trained Multilingual ASR Model

A large pre-trained language model, also called a foundation model, is a machine learning model that can perform various general-purpose natural language processing (NLP) tasks. The model is trained with a vast amount of unlabelled data and by leveraging deep learning techniques, particularly deep neural networks (DNN) and transformer architectures, the model can capture complex patterns from massive training data and therefore can understand and generate human language. Over the years, the pre-trained model size has grown larger and larger. In 2018, the model size of the GPT (Radford et al., 2018) had only 117 million parameters, whereas in 2020 the GPT-3 (Brown et al., 2020) had 175 billion parameters. Scaling up the model accompanied the significant improvements in model performance for few-shot settings, and can achieve state-of-the-art fine-tuning performance of prior models (Brown et al., 2020).

In the speech domain, there are several large pre-trained multilingual ASR (MASR) models that have remarkable performance in recognizing multiple languages. Such models are generally trained with huge amounts of multilingual speech data. The most well-known MASR models are XLS-R (Babu et al., 2021), Whisper (Radford et al., 2022b), USM (Zhang et al., 2023) and MMS (Pratap et al., 2023). Proposed by Facebook, XLS-R, containing over 2 billion parameters, is built on wav2vec 2.0 and is able to recognize 128 languages. Whisper, built by OpenAI, has the capability to transcribe 99 languages with 1.55 billion parameters. Recently, the Google USM model with 2 billion parameters was created to perform speech recognition tasks across more than 100 languages. The MMS built by Meta AI with 1 billion parameters is able to recognize over 1000 languages.

MASR has shown a great potential in speech recognition tasks, not only for its powerful capability to recognize multiple languages but also for its better performance than monolingual models baselines for many low-resource languages such as Persian, Telugu, Esperanto, Kyrgyz, and so on (Hou et al., 2020; Pratap, Sriram, et al., 2020). It has been suggested that the improved performance of low-resource languages in MASR comes from the data pooling and transfer learning from similar languages, and cross-language joint optimization (B. Li et al., 2021).

However, one huge pitfall of large pre-trained multilingual ASR models is that their performance in low-resource languages is still poor, compared with high-resource languages. For example, with less than 5% WER for recognizing high-resource languages such as English or Spanish, Whisper performs poorly on LRLs such as Hindi or Icelandic and so on with a higher than 30%

WER. Several low languages with the worst performance have WER between 80% to 160%. There is a clear linear relationship between the size of pre-trained language data and the WER of that language. These LRLs, taking up an extremely small proportion of the training data of the Whisper model, have the worst WER performance. Among 680k hours of training data, Icelandic only has 16 hours and Hindi has 12 hours.

2.1.1 Whisper Model

Whisper is by far one of the most powerful large pre-trained multilingual ASR models. It is a large-scale weakly supervised pre-training model trained on 680k hours of labeled speech audio. Of these 680k hours of speech data, 117k hours are audio from 96 languages other than English and 125k hours are X → English translation data. Such data distribution enables multilingual and multitask training for Whisper. The transcription data Whisper used to train are Internet text resources paired with its corresponding audio, which means the quality of transcription is diverse. The authors do not bother to perform any gold-standard human validation or significant text normalization. Instead, they rely on the naturalness and expressiveness of the original transcripts. One important step they work on the transcripts is that they remove the machine-generated transcripts, partially transcribed transcripts and wrongly aligned transcripts.

The Whisper model has encoder-decoder transformer architecture. The architecture is shown in Figure 1. All audios are first broken down into 30-second segments and are paired with the transcript for within these segments. Audio segments are then resampled to 16kHz and Mel spectrogram representation is computed. The model takes the Mel spectrogram as input and passes the input to two 1-D convolutional layers and the GELU activation function. Then sinusoidal position embeddings are added to the output after which the output is passed to transformer encoder blocks. Each block is a pre-activation residual block consisting of MLP layers and self-attention layers. A final normalization layer is applied to the encoder output before passing the output to the decoder blocks. The decoder uses learned position embeddings and tied input-output token representations.

Whisper shows remarkable performance in English and other languages. The best zero-shot Whisper model has a LibriSpeech clean-test WER of 2.5%, which is a comparable result with the best supervised LibriSpeech model. When testing on other datasets, Whisper tremendously outperforms all benchmarks from LibriSpeech models. The tiniest zero-shot Whisper model that only has 39 million parameters can achieve close results as supervised LibriSpeech models when testing on other datasets. Overall, zero-shot Whisper achieves a 55.2% WER reduction for English on other ASR datasets than the supervised model Wav2Vec 2.0. In terms of multilingual speech recognition capabilities, results on several multilingual datasets including Multilingual LibriSpeech (Pratap, Xu, et al., 2020), VoxPopuli (Wang et al., 2021), and Fleurs (Conneau et al., 2022) are reported. Zero-shot Whisper outperformed XLS-R (Babu et al., 2021), mSLAM (Bapna et al., 2022), and Maestro (Z. Chen et al., 2022) on 15 languages in Multilingual LibriSpeech. Depending on the amount of training data for languages, Zero-shot Whisper is able to recognize 72 languages in Fleurs with varying WER from 2.5% to 160%.

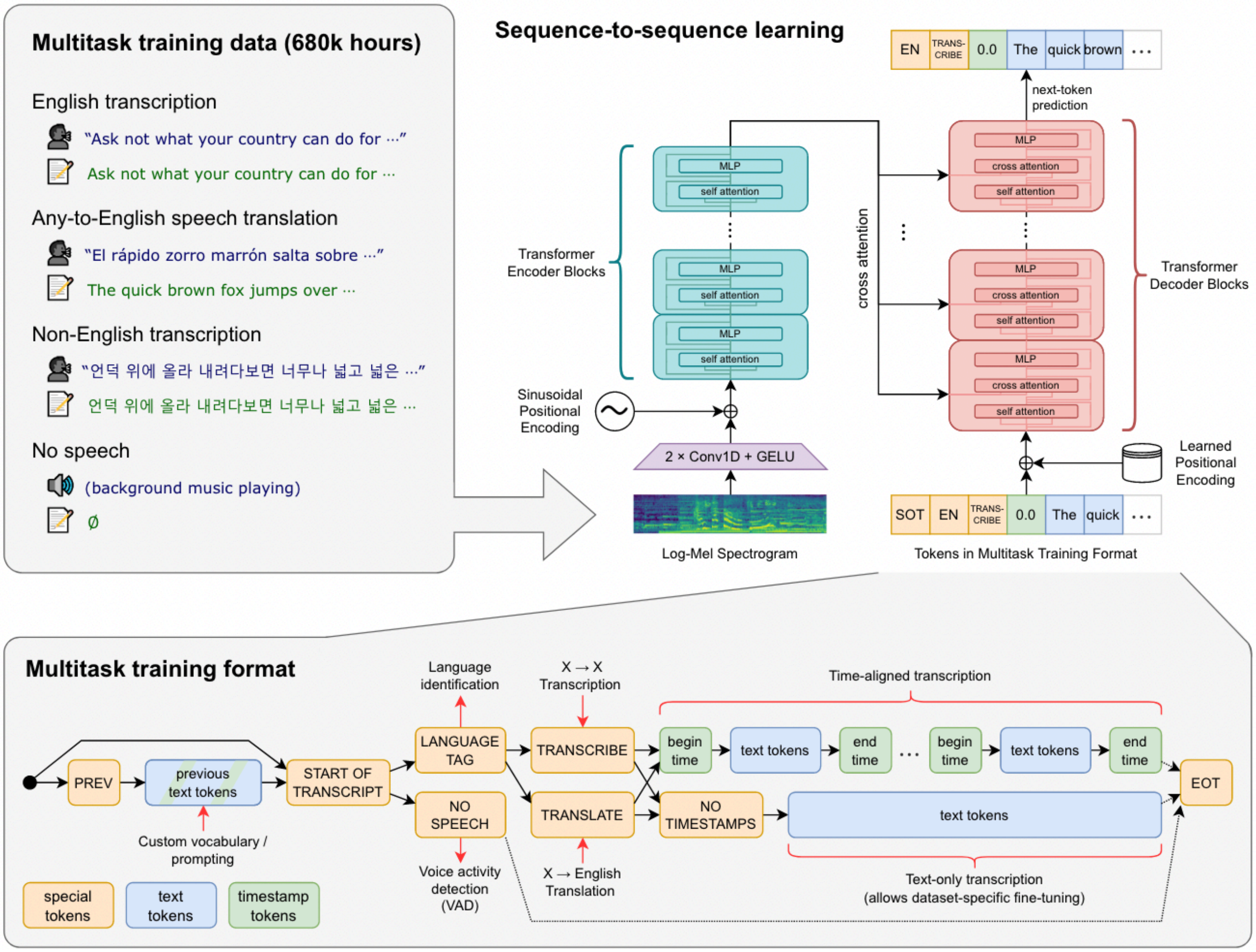

Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ⋯"
Ask not what your country can do for ⋯
Any-to-English speech translation
"El rápido zorro marrón salta sobre ⋯"
The quick brown fox jumps over ⋯
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
⋯
next-token prediction
MLP
cross attention
self attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
LANGUAGE TAG
TRANSCRIBE
begin time
text tokens
end time
PREV
previous text tokens
START OF TRANSCRIPT
NO SPEECH
TRANSLATE
NO TIMESTAMPS
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Figure 1. The architecture of the Whisper model (Radford et al., 2022b).

Model Size	Number of Parameters	English-only	Multilingual
tiny	39 M	Yes	Yes
base	74 M	Yes	Yes
small	244 M	Yes	Yes
medium	769 M	Yes	Yes
large	1550 M	No	Yes
large-v2	1550 M	No	Yes
large-v3	1550 M	No	Yes

Table 1. The summarization of current Whisper checkpoints.

Currently, Whisper has seven checkpoints available. The smallest checkpoint has 39 million parameters, and the largest checkpoint has 1550 million parameters. The different checkpoints have been summarized in Table 1. All of the checkpoints support multilingual speech recognition, and only the smallest four have English monolingual checkpoints. The performance differs based on the size of the models. The larger the model, the better the performance. As seen in Hugging Face open ASR leaderboard[1], the English monolingual Whisper medium model (769M) outperforms the English monolingual Whisper small model (244M), with an average WER 8.5% and WER 9.34% respectively. The largest version of Whisper has an average WER of 7.7% across several English datasets including Common Voice 9, Librispeech, and Voxpopuli, ranking 6th place on the Hugging Face open ASR leaderboard.

2.1.2 Full Fine-Tuning of Large-Pretrained Model

As discussed in the earlier section, large pre-trained models are trained on massive amounts of unlabelled data that enable them to perform general-purpose tasks, but they might not be good at downstream domain-specific tasks. To improve the performance of such downstream tasks, the pre-trained base model needs to be continuously trained on a newer, smaller dataset from downstream tasks. This approach is called full fine-tuning. For example, a self-supervised pre-trained ASR model wav2vec 2.0 that is good at performing general speech recognition tasks could be further fully fine-tuned on accented speech data for accent identification tasks and accented speech recognition tasks (Deng et al., 2021). During full fine-tuning, the base model is first initialized with the pre-trained model parameters, and all of the parameters are re-trained with labeled data from downstream tasks. This way a downstream task model will not be trained from scratch which greatly reduces training time and resources, and it leverages the profound knowledge of a pre-trained model.

[1] https://huggingface.co/spaces/hf-audio/open_asr_leaderboard

Full fine-tuning has shown powerful adaptation ability on pre-trained monolingual ASR models for low-resource languages. (Yi et al., 2020) fully fine-tuned the pre-trained wav2vec 2.0 model, which is a monolingual English model, to solve low-resource language recognition tasks. In the experiment, wav2vec2.0 was used as an encoder. Six languages including Mandarin, English, Japanese, Arabic, German, and Spanish were selected as training languages. Though theoretically, these languages are not low-resource languages, given the fact that these languages (except English) are unseen languages to the wav2vec 2.0 model and only 15 hours of training data for each language are used for training, this simulates the low-resource scenarios. The results showed that full fine-tuning has successfully adapted the pre-trained wav2vec2.0 model to recognize five additional languages and achieved more than 20% improvement than non-pretrained models.

Full fine-tuning techniques have also been applied to pre-trained multilingual ASR models. (Javed et al., 2022) built a multilingual wav2vec model using 40 Indian languages and 17000 hours of raw speech data. They then fully fine-tuned their model for 9 languages for downstream tasks and achieved state-of-the-art results for very low-resource languages such as Nepali and Sinhala. Similarly, (Zhao & Zhang, 2022) fully fine-tuned XLSR-53 models with 10 hours of labeled training data for 10 selected low-resource languages including Tagalog, Swahili, Javanese, and so on. They found that simply fully fine-tuning a pre-trained multilingual model performed better than traditional hybrid Deep Neural Network (DNN) or Hidden Markov Model (HMM) architecture for low-resource languages.

As for the recently developed multilingual pre-trained ASR model Whisper, there is very little research on fully fine-tuning it for low-resource languages. (Williams et al., 2023) recently explored the applicability of fine-tuning Whisper for Maltese ASR. Different sizes of Whisper models were fully fine-tuned with Maltese training data ranging from as little as 10 minutes to as much as 50 hours. They evaluated fully fine-tuned model performance with testing data from MASRI (Mena et al., 2020) and Common Voice. The results showed that the WER of Maltese on Whisper models dropped from 100% to below 40% as the size of training data increases. (Rouditchenko et al., 2023) expanded the Whisper's ability to adapt to 13 unseen and 18 seen languages by fully fine-tuning it with target languages. Unseen means that these languages are not part of the pre-trained languages for Whisper, and among 18 seen languages some of them are low-resource languages such as Maltese, Bengali, and so on. The authors used only 12 hours of training data from FLEURS for each language. After full fine-tuning, these low-resource languages saw a huge WER reduction from above 100% to less than 10%. Overall, full fine-tuning research on multilingual pre-trained ASR models often showed great benefits for low-resource languages.

Despite the remarkable benefits full fine-tuning brings to the model, fully fine-tuning a pre-trained model could be very costly. Since all parameters in the pre-trained model are re-trained for the new downstream task, the training process is time-consuming and resource-inefficient. Meanwhile, it requires large storage space (Z.-C. Chen et al., 2023) to save the downstream model which has about the same model size as the original pre-trained large model. Besides, overwriting the original parameters is not an efficient way to use the shared knowledge of pre-trained models (Z.-C. Chen et al., 2023).

2.2 Parameter-Efficient Fine-Tuning (PEFT)

In the previous chapter, I talked about the benefits as well as downsides of the full fine-tuning method. In this chapter, I introduce the solution to mitigate the heavy computational cost of full fine-tuning: Parameter-Efficient Fine-Tuning (PEFT), an efficient way to train the model for downstream tasks. This approach aims to reduce the number of trainable parameters while maintaining comparable performance to the full fine-tuning by freezing the backbone of a large pre-trained model and only updating a small number of parameters.

Based on research by (Xu et al., 2023), as of 2023, there are more than 50 different PEFT approaches. These approaches can be divided into five categories: additive fine-tuning, reparameterized fine-tuning, hybrid fine-tuning, partial fine-tuning, and unified fine-tuning. Under these categories, there are even detailed classifications such as adapter-based fine-tuning, Bias update, Low-rank decomposition, Manual combination, and so on. In this part, I introduce one method for each category and their usage in speech-related research.

2.2.1 Additive Fine-Tuning

Adapter-based fine-tuning is a category under additive fine-tuning. The name of this category is self-explanatory, implying that adapter modules are additively added to the model. The idea of an adapter is first introduced by (Rebuffi et al., 2017) in the domain of computer vision. Specifically, they proposed a residual adapter module that contained only less than 10% of overall parameters and was able to perform a high-degree parameter sharing among domains. Inspired by their work, (Houlsby et al., 2019) introduced a bottleneck adapter to handle the large parameters in the NLP domain.

As shown in Figure 2, a typical bottleneck adapter consists of a down-projection matrix W_{down}, a nonlinearity f , an up-projection matrix W_{up} , and a residual connection r . Each activation is projected down to a smaller dimensionality and then passes through a nonlinear function. The result will then be projected back up to the original dimension. Last, a residual connection is added to the result before passing the result to the next layer. The formula for this process is shown as follows:

$$h = W_{up} \cdot f(W_{down} \cdot h) + r$$

This process ensures that new extra trainable parameters are introduced for a specific task and the pretrained parameters are not modified.

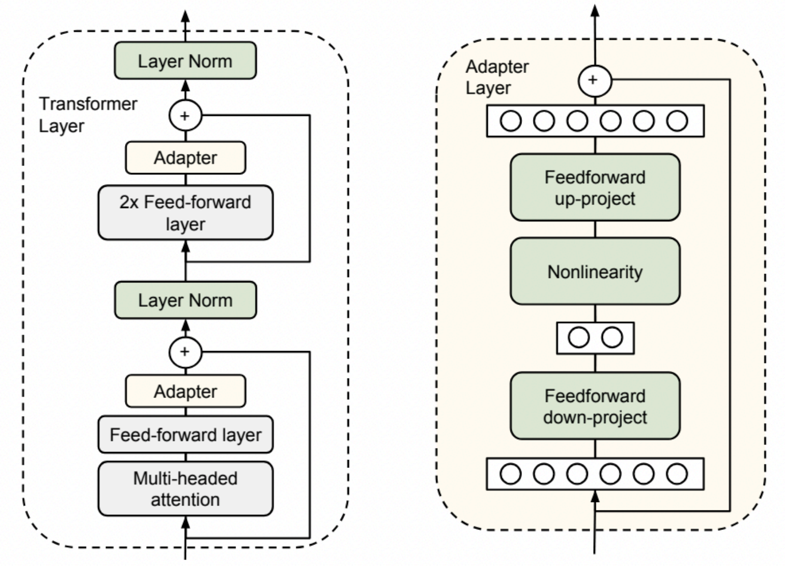


Figure 2. The structure of adapters (Houlsby et al., 2019).

The adapter-based approaches have been quickly adopted in the ASR domain to perform speech recognition tasks. In 2019, (Kannan et al., 2019) proposed to use bottleneck adapters on top of the multilingual RNN-T model to further improve the performance of 9 Indian languages. A multilingual RNN-T base model was trained on the data of all languages, in their case, 9 Indian languages. Then all model parameters were frozen and adapter modules were introduced after every layer of the encoder. Each language had its own adapter modules with separate parameters. At the inference stage, only the language-specific adapter was applied. They found that with only 2% of the original model size, the multilingual model with the adapter was able to further improve the language performance and even outperformed the baseline monolingual models.

Inspired by Kannan et al, later research employed adapters on a variety of models for all kinds of downstream ASR tasks. For example, adapters have been used on RNN-T and T-T (Transformer Transducers) models to recognize pathological speech and heavily accented speech (Tomanek et al., 2021). Adapters were also used for language identification tasks (e.g. Arabic dialects) on Whisper (Radhakrishnan et al., 2023) and code-switching tasks (e.g. Mandarin-English) on pre-trained wav2vec2.0 model (C.-Y. He & Chien, 2023). All of their results showed that by updating only a tiny fraction of the model parameters, the model performed very well on downstream ASR tasks.

2.2.2 Reparameterized Fine-Tuning

Though the previously mentioned PEFT method (i.e. adapters) is less computationally expensive, it has a downside by increasing the inference latency and hard to maintain comparable performance as full fine-tuning baselines, as suggested by (Hu et al., 2021). The reason is, in the adapter approach, despite a few small adapter layers being added into the pre-trained model, the model inputs still use all parameters during inference, which makes the inference time the same or even slower compared to the full fine-tuning approach.

Motivated by the disadvantages of existing adapter approaches, (Hu et al., 2021) proposed Low Rank Adaptation (LoRA), aiming to boost the efficiency of adapting large language models to downstream tasks during both training and inference. The structure of LoRA is shown in Figure 3. Two trainable low-rank decomposition matrices are inserted into the layers of a pre-trained model. The formula is as follows:

$$h = W_0 x + \frac{\alpha}{r} BAx$$

$W_0 \in R^{d\times k}$ is the pre-trained matrix. $A \in R^{r\times k}$ and $B \in R^{d\times r}$ are two decomposition matrices that contain trainable weights. The rank is $r \ll min(d, k)$ and α is a constant in r. The update of the pre-trained matrix W_0 is therefore constrained by low-rank decomposition matrices A and B. During training, W_0 is frozen and is not updated, while A and B are updated accordingly. A is initialized as a random Gaussian and B is initialized as zero and then BAx is scaled by $\frac{\alpha}{r}$. r is a pivotal hyperparameter in LoRA, because the size of r determines its performance. The smaller the r is, the faster and more cost efficient the training process. Large r value hinders the computational efficiency but allows the model to handle complex tasks. It is important to note that the learned weights of A and B can be merged with the main weights W_0, which means for different downstream tasks W_0 can be recovered by subtracting BA and adding a new $B^{'}A^{'}$, therefore introducing no additional latency during inference.

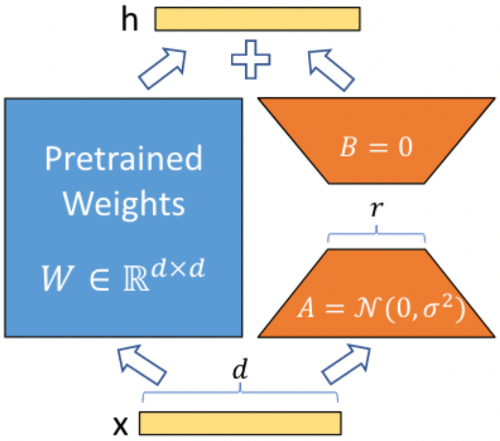

h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 3. The structure of LoRA (Hu et al., 2021).

In their experiments, for simplicity, they only inserted LoRA to W_q and W_v which is a matrix for query and value, respectively. They then evaluated the downstream task performance including

NLU (natural language understanding) and NLG (natural language generation) of LoRA on RoBERTa, DeBERTa, GPT-2, and GPT-3, and compared the results with full fine-tuning and different types of adapters. They found that LoRA outperformed many baselines with comparable or even smaller numbers of training parameters.

The LoRA approach has been widely used in the domain of speech for a variety of downstream tasks including speech recognition and emotion recognition, thanks to the great benefits that it reduces the number of parameters during training while not increasing the inference time. For example, (Liu et al., 2024) applied LoRA to the Whisper model and improved the performance of recognizing Chinese child speech. (Feng & Narayanan, 2023) inserted LoRA on Whisper models, wav2vec 2.0 base, and WavLM base (S. Chen et al., 2022), and found out these pre-trained models can be successfully adapted to recognizing speech emotions, and LoRA outperformed other PEFT techniques in this task.

2.2.3 Partial Fine-Tuning

The partial fine-tuning technique reduces the number of trainable parameters by only updating the subset of parameters that are important to downstream tasks. One of the categorizations in partial fine-tuning is bias update which only updates the bias terms of the transformer, for example, Bit-Fit (Zaken et al., 2021).

Bit-Fit (BIas-Term FIne-Tuning) was first proposed by (Zaken et al., 2021) to reduce the trainable parameters of the large language model BERT. The idea behind this method is to freeze most of the transformer-encoder parameters and only train bias term and task-specific classification layers. In their experiment, they tuned bias terms from key, query, and value encoders of self-attention heads in attention layers and bias terms in feed-forward and layer normalization layers. These bias terms only make up 0.09% and 0.08% of the total number of parameters in the BERT base and BERT large model. They also tried to only tune a subset of bias parameters in the query and the second MLP layer. The results showed that this approach maintained good performance on all BLUE tasks.

There is very little research investigating Bit-Fit for ASR. (Z.-C. Chen et al., 2023) investigated the effectiveness of several PEFT methods including Bit-Fit on self-supervised speech models such as wav2vec 2.0 for downstream tasks such as ASR and phoneme recognition (PR). In their experiment of Bit-Fit, instead of tuning the bias term, they tuned the weights of all modules in the model. All PEFT modules were trained using 1 hour and 10 hours of training data from Libri-Light to simulate low-resource scenarios and tested the model on a test set of LibriSpeech. Their results showed that Bit-Fit performed better than full fine-tuning when the training data is scarce, but among all kinds of PEFT methods, the Bit-Fit approach had an average performance. Bit-Fit only had the best performance for the keyword spotting task.

2.2.4 Hybrid Fine-Tuning

Since the PEFT methods show great benefits in adapting large models to downstream tasks, researchers decided to explore if the combination of different PEFT methods can be more effective, hence hybrid. One of the hybrid fine-tuning methods is the Mix-and-Match adapter (MAM), proposed by (J. He et al., 2021). This approach combines prefix tuning and parallel adapters. More specifically, the authors used a small bottleneck dimension for prefix tuning at attention sub-layers and allocated more parameter budgets to modify the representation of the feed-forward network (FFN) using the scaled parallel adapter. The parallel adapter computes representations in parallel to the transformer feed-forward layers or attention layers.

The only study using a MAM in the speech domain was done by (Peng et al., 2022). Three pretrained speech models HuBERT base, WavLM base, and WavLM large were adapted using a MAM adapter for speaker verification tasks on the VoxCeleb corpus (Nagrani et al., 2017). By only updating less than 4% of parameters of the original model parameters, the MAM adapter was able to achieve comparable performance to the full fine-tuning in low-resource scenarios.

2.2.5 Unified Fine-Tuning

The unified fine-tuning aims to streamline the adaptation and optimization of a model (Xu et al., 2023). Unlike the combination of PEFT methods in hybrid fine-tuning, unified fine-tuning typically only involves one PEFT method. One of the unified fine-tuning methods is the Sparse Adapter, proposed by (S. He et al., 2022). They argued that the existing adapter methods have to increase the overall model parameters to match the performance of full fine-tuning. Therefore, they proposed to prune the adapter before using it for tuning. The method works as follows: first a target sparsity s is set and a score z is assigned to all parameters w. The threshold z_s, i.e. the s-th percentile of z is computed. The parameters whose scores are below z_s are considered redundant and will be removed. This plug-in method (Xu et al., 2023) can be applied to a variety of PEFT methods including LoRA, Adapter, MAD-X (Pfeiffer, Vulić, et al., 2020), MAM adapter, and AdapterFusion (Pfeiffer, Kamath, et al., 2020). The results showed that with 40% sparsity, the Sparse Adapter still outperformed its baseline models that did not apply the Sparse Adapter. So far, Sparse Adapter has not been used in any ASR research.

2.3 PEFT for Pre-Trained Multilingual ASR Models

Combining previous discussions on pre-trained large multilingual ASR models and fine-tuning, now it should be clear that MASR is good at recognizing multiple languages, but its performance on downstream tasks, such as recognizing low-resource languages is still undesired. A typical way to improve the performance of a pre-trained ASR model is through fully fine-tuning the entire model with additional data of target language, but this approach is computationally expensive due to the huge size of model parameters. The PEFT method is therefore developed to mitigate this problem, aiming to reduce the number of trainable parameters while retaining the comparable model performance as the full fine-tuning method.

In the last chapter, I gave a broad overview of a variety of PEFT methods and their usage in the speech domain. In this chapter, I further narrow down the scope of the PEFT study to pre-trained multilingual ASR models, focusing on how PEFT is used to adapt MASR models to low-resource languages. I also reviewed several articles about English in this section, though English is not a low-resource language. The reason is that English models will be used as a baseline in my experiments, so it is important to know the LoRA performance for English from the literature.

2.3.1 LoRA for English

In (Fathullah et al., 2023)'s study, the authors explored whether decoder-only large language models can be enabled with speech recognition ability by conditioning on audio embeddings. They inserted LoRA modules on key, query, value, and output layers of the smallest LLaMA-7B model and trained LoRA with Multilingual LibriSpeech (MLS) datasets. By default, r was set to 8 and α was set to 16. They found that LoRA successfully adapted decoder-only LLaMA models to perform English speech recognition tasks and the model for several languages such as Dutch, French even outperformed its monolingual baselines.

(Southwell et al., 2024) conducted experiments on Whisper to improve ASR performance for English child speech in classroom environments. They used LoRA to tune the Whisper large-v2 and Whisper base model. To further reduce the number of parameters, they also used int8 quantization. In addition, they fully fine-tuned the Whisper base model. They used rank $r = 32$, scaling factor $\alpha = 64$, dropout = 0.05, learning rate = 1e-4, 50 warmup steps, and a linear decay. The results showed that LoRA reduced WER for both large and base Whisper models for English child speech, suggesting that LoRA improved the model performance.

2.3.2 LoRA for Low-Resource Languages

(Kim et al., 2023) is the first research about inserting LoRA modules into a multilingual Whisper model to adapt Whisper to low-resource language situations. They first inserted LoRA modules into the attention heads of decoder layers, followed by pruning 50% of the Whisper large model parameters using the Lottery Ticket Hypothesis (LTH). They also experimented with fully fine-tuning the model. Six language datasets from the Common Voice corpus, i.e. English, Chinese, Korean, Malayalam, Japanese, and Swahili were used for training and testing. The size of training and testing sets varied in size, from only 192 sentences in the Korean training set to 29383 sentences in the Chinese training set. They found that using 1.5B parameters, fully fine-tuning the model achieved the best WER for all six languages. Meanwhile, with merely 2.6M parameters, LoRA could achieve comparable CER and WER for all six languages to the full fine-tuning approach. Such results indicated that LoRA was very effective in adapting the Whisper model to low-resource languages. However, arguably, only Malayalam and Swahili could be considered as low-resource languages because each of the other languages have already been pre-trained with more than 7000 hours of data in Whisper.

Another study by (Ferraz et al., 2024a) also applied LoRA to the Whisper model under a low-resource language context. They proposed a new model distill-whisper, which unlike the current Distil-Whisper models on HuggingFace[2], their model maintained Whisper's multilingual capabilities. In other words, their distilled model can recognize languages other than English. To evaluate the model performance, they compared their model to a fully fine-tuned Whisper small model, a LoRA-tuned Whisper small model and a Whisper small model. In their experiment, they inserted LoRA on top of the feed-forward layers. They trained the model with 8 languages Catalan, Czech, Galician, Hungarian, Polish, Thai, Tamil, and Ukrainian using only 14 hours of training data for each language from Common Voice 13 (CV-13) datasets, and tested the performance on both CV-13 and FLEURS datasets. They found out that LoRA increased the performance of both CV-13 and FLEURS datasets compared with the base model Whisper small, and LoRA also achieved comparable performance to full fine-tuning.

(Do et al., 2023) applied high-rank LoRA with $r = 192$ on the Whisper tiny model and trained the model with Vietnamese speech. They inserted LoRA on all linear layers within the model, including query, key, value, output projection, MLP, and E_{out}. It is worth noting that LoRA was implemented together with the decoupling of token embeddings, meaning the model learns input and output token representation separately. They compared the result with full fine-tuning and zero-shot setting of Whisper tiny and found that high-rank LoRA yielded comparable performance improvement as to full fine-tuning. However, since the Whisper model architecture is changed when applying LoRA, it is difficult to tell whether the improvement of WER comes from LoRA or the new model architectures.

To sum up, the previous research shows that LoRA can be very effective in adapting Whisper to low-resource languages. However, the low-resource languages in previous research are all supported by Whisper, which means that Whisper has seen the languages in its pre-training data. Some of the so-called low-resource languages in the literature, such as Vietnamese, Korean, Ukrainian, and so on are theoretically not that low-resourced for Whisper, given that they have at least hundreds of pre-training data in Whisper. Besides, many other low-resource languages largely remain unexplored. Therefore, it is still unknown if a low-resource language that has only hours or dozens of pre-training data in Whisper, or even an unseen language in Whisper, for example, Frisian, could also benefit from LoRA.

2.4 Frisian ASR

Frisian is a West Germanic language spoken by 400,000 Frisian people and there are three variants of Frisian: West Frisian, North Frisian, and East Frisian. West Frisian is by far the most

[2] https://huggingface.co/distil-whisper

spoken of the three variants and has been officially recognized as the second language of the Netherlands in Fryslân. The Frisian has also been recognized as a minority language.

The first Frisian ASR study was conducted by (Yılmaz et al., 2016). In the study, they built a language-dependent and language-independent bilingual deep neural network-based ASR model to recognize code-switching speech between Frisian and Dutch. The dataset they used was a Frisian corpus called FAME!, a Frisian Radio Broadcast Database designed for code-switching study by (Yilmaz et al., 2016). They achieved WER 36.4% for the language-independent model and WER 36.3% for the language-dependent model.

The state-of-the-art Frisian ASR model was built by (Bălan, 2023)[3]. In his work, using 41 hours of Frisian data from the Common Voice 8.0 dataset, he fully fine-tuned the XLS-R model, a large-scale cross-lingual pre-trained model. His approach achieved 4.11% WER and set the new benchmark for the Frisian ASR model. Before this new benchmark was developed, (de Vries, 2021)[4] and (Crang, 2021)[5] attempted to achieve Frisian WER 16.25% and WER 19.11% on XLSR-53, respectively.

[3] https://campus-fryslan.studenttheses.ub.rug.nl/360/1/MA%20S3944867%20DA%20Balan.pdf
[4] https://huggingface.co/wietsedv/wav2vec2-large-xlsr-53-frisian
[5] https://huggingface.co/crang/wav2vec2-large-xlsr-53-frisian

3 Methodology

In this section, I will introduce the methodology for the study, by first introducing the datasets used in the study and addressing the importance of data processing. In 3.2, I simply introduce the model chosen for the study and in 3.3, the evaluation metrics for model performance will be elaborated. In 3.4, detailed experimental set-ups will be explained.

3.1 Datasets

Common Voice is a publicly accessible multilingual speech corpus created by Mozilla (Ardila et al., 2019). It aims to create a free database for speech recognition software. The first database was released in 2017 and as of 2022, there are more than 100 languages in the database with more than 30000 hours of recording. As common voice is a crowdsourcing project, anyone can donate their voice to the database and validate other people's speech clips.

In this study, the Frisian dataset from Common Voice Corpus 6.1, which is released in 2020, is used. Though the dataset contains 47 hours of recordings, only 15 hours are validated. To ensure the quality of the data, I only use the 15-hour validation split for training and testing. Around 5 hours of data is split for testing and the rest of 10 hours of data is used for training. Among the 10 hours of training data, 10 minutes and 1 hour of recordings are further extracted for the experiments on the effect of training data size. The speakers in these datasets vary across different age groups and gender groups. All audios are in mp3 format and the sampling rate is at 32kHz, which will be later converted to 16kHz during the data preprocessing phase. Although the most recent Frisian datasets in Common Voice Corpus 16.1, which has around 69 hours of validated data, I decided to use an earlier version of Common Voice dataset with less validated data. The reason is that data scarcity is a common problem for many low-resource languages, and 15 hours of data is closer to a real-life low-resource situation.

Another popular Frisian dataset is FAME!, which stands for Frisian Audio Mining Enterprise. It contains 18.5 hours of annotated radio broadcasts in the Frisian language. Despite its high quality, it is a bilingual database, aiming for code-switching research between Frisian and Dutch, which means this dataset includes lots of Dutch words and sentences. The bilingual feature makes this corpus less favorable for my study. Additionally, this corpus is not easily accessible because it requires the legal authorisation of two parties before permitting to download. Taking all these into consideration, I decided to use the Common Voice Corpus dataset for the current study.

English is also evaluated in the study mainly as a baseline. As for English, LibriSpeech ASR corpus (Panayotov et al., 2015) is used in experiments. LibriSpeech is an open-source corpus containing 1000 hours of read English speech sampled at 16 kHz and the speech data is all derived from audiobooks from the LibriVox project. The speech data in LibriSpeech are all aligned and segmented with a high quality. In the study, 10 hours of speech data extracted from train-clean-100 split are used for training, and 5 hours of speech data extracted from the test-clean split are used for testing. In addition, 10 minutes and 1 hour of speech data are also extracted from the train-clean-100 split for the experiments on the effect of training data size. The reason for using the Librispeech dataset is that it is so commonly used in ASR research that I can get enough benchmark results for comparison. Another advantage of using the Librispeech dataset is that the dataset has very high quality compared with crowdsourcing collected Common Voice corpus. The dataset information is summarized in Table 2.

Datasets	Frisian	English
	Common Voice 6.1	LibriSpeech
Training (~10 mins/ ~1 hour/ ~10 hours)	Validation	train-clean-100
Testing (~5 hours)	Validation	test-clean

Table 2. The summarization of training and testing datasets in the study. Common Voice 6.1 validation split is used for Frisian. LibriSpeech train-clean-100 split and test-clean split are used for English. The training datasets vary in size, ranging from 10 minutes, 1 hour to 10 hours. The testing dataset has around 5 hours of speech data.

In this section, I also emphasize the importance of data normalization for achieving accurate WER results for both Frisian and English. In LibriSpeech corpus, all transcriptions are in capital letters, for example, “ AND JUNE EIGHTEEN FORTY EIGHT KNEW A GREAT DEAL MORE ABOUT IT THAN JUNE EIGHTEEN THIRTY TWO SO THE BARRICADE OF THE”. This reference sentence will be recognized by Whisper as “June 1848 knew a great deal more about it than June 1832. So the barricade of the”. The WER calculated by jiwer package of this sentence is 1, meaning the entire sentence is wrongly transcribed, which is not the case. It is clear that the WER calculator is very sensitive to cases and does not understand “EIGHTEEN FORTY EIGHT” and 1848 refer to the same year. Two possible solutions to address the problem are, first, converting sentences to lowercase and second, normalizing sentences.

I tested two solutions using the above-mentioned sentence with the Whisper small model. By only converting the reference sentence to lowercase, the WER is 45.45%. By converting both the reference sentence and Whisper's predicted sentence, the WER of 31.82%. Both results far lag behind the reported Whisper performance on LibriSpeech English (Radford et al., 2022c). I then tested the normalization solution, which is to remove all punctuations and symbols, restore letters to numbers (e.g. “EIGHTEEN THIRTY TWO” to “1832”), and convert all letters to lowercase. This can be done by using the English text normalizer from the Whisper normalizer package. By normalizing the reference sentence, the WER is 27.78%, whereas by normalizing both reference and prediction, the WER is 5.56%, which is more reasonable for the performance of Whisper for English. Therefore, for all experiments, both reference sentences and predicted sentences will be normalized before calculating WER.

For the same reason that is to avoid miscalculation of WER in Frisian, all Frisian reference sentences and predicted sentences will be normalized by a basic text normalizer from the Whisper normalizer package before calculating WER.

3.2 Models

In this study, I particularly explore the 244M Whisper-small model. This decision is made considering the trade-off between model size and model performance. The checkpoint should be small enough to run on a single A100 GPU with relatively good performance.

3.3 Evaluation Metrics

To evaluate the performance of LoRA and full fine-tuning, several evaluation metrics are introduced to the study. The first one is the most frequently used metric in ASR systems, namely, word error rate (WER). It measures the accuracy of transcribed text compared to a ground truth text. The lower the WER, the more accurate the transcription, hence better model performance. The WER is calculated by taking into account the number of substitutions, insertions, and deletions. Substitutions are incorrect words. Insertions are words that are not in the ground truth text. Deletions are words that are missing. The following formula shows how WER is calculated:

$$Word\ Error\ Rate\ = \frac{Substitutions + Insertions + Deletions}{Number\ of\ Words\ Spoken}$$

The second metric is the number of trainable parameters. When tuning a pre-trained model to downstream tasks, parameters in the model need to be retrained. The more parameters that are involved in this retraining process, the more computationally expensive this process is. To achieve efficiency, only a small number of parameters should be retrained.

The third metric is GPU memory usage during the training. This information is retrieved by running “nvidia-smi” on the GPU node while each experiment is running. In this way, the amount of GPU memory used for the current training can be acquired.

3.4 Experiments

All experiments are conducted on Habrok, a computer cluster provided by the University of Groningen. I requested an Nvidia A100 GPU with 40GB of VRAM. The training details of all experiments are summarized in Table 3.

3.4.1 Zero-Shot Evaluation

It has been suggested that Whisper has the reliable ability to generalize well across languages without being fine-tuned by language-specific data (Radford et al., 2022b). In this experiment, I tested the generalization capability of Whisper for Frisian and English, and the results served as baselines for other experiments. The procedure of the zero-shot experiments is as follows: using the Whisper pipeline on HuggingFace, 5 hours of Frisian testing dataset and 5 hours of English testing dataset are transcribed, and WER is calculated for each testing dataset. The batch size during evaluation is 16 and the audio is chunked into 30 seconds.

3.4.2 LoRA Experiments

Six LoRA experiments were conducted in the study. The hugging Face PEFT library was used for a faster and easier implementation of LoRA. Following the methods of (Hu et al., 2021), LoRA were inserted into value and query projection matrices in the self-attention module, because the authors suggested that adapting both value and query projection matrices yielded the best performance. r was set to 32, because after trying different settings ranging from 2 to 64, it seems 32 gave the best trade-off between a number of trainable parameters and model performance. α was set to 64 and lora dropout was 0.05. Bisas is none.

All Frisian models were trained using learning rate 1e-3 with training batch size 8, evaluation batch size 8, 50 steps warm-up, 1 gradient accumulation step, Adam optimizer with 1e-08 epsilon, seed 42, and epoch as evaluation strategy. The 10mins Frisian model was trained for 10 epochs and another two models were trained for 15 epochs. Since Frisian is not a supported language in Whisper, the language in the whisper tokenizer and whisper processor were all set to Dutch, a closely related language to Frisian.

As for English models, the 10-mins model was trained for 40 epochs with a learning rate of 1e-3. The 1-hour model was trained for 30 epochs with a learning rate of 1e-4, and the 10-hour model was trained for 10 epochs with a learning rate of 1e-5. They were all trained with batch size 8, evaluation batch size 8, 50 steps warm-up, 1 gradient accumulation step, Adam optimizer with 1e-08 epsilon, seed 42, and epoch as evaluation strategy.

	Frisian					
	LoRA			Full Fine-Tuning		
Models	10mins	1 hour	10 hours	10mins	1 hour	10 hours
Learning Rate	1e-3	1e-3	1e-3	1e-5	1e-5	1e-5
Training Batch Size	8	8	8	8	8	8
Evaluation Batch Size	8	8	8	8	8	8
Epoch	10	15	15	16.6	11.2	1.6
Steps	150	1335	14025	250	1000	1500

(a) The training hyperparameters for Frisian

	English					
	LoRA			Full Fine-Tuning		
Models	10mins	1 hour	10 hours	10mins	1 hour	10 hours
Learning Rate	1e-3	1e-4	1e-5	5e-7	5e-7	5e-7
Training Batch Size	8	8	8	8	8	8
Evaluation Batch Size	8	8	8	8	8	8
Epoch	40	30	10	333.3	51.2	5.6
Steps	240	1170	3600	1000	1000	1000

(b) The training hyperparameters for English

	Training Time					
	Frisian			English		
Models	10mins	1 hour	10 hours	10mins	1 hour	10 hours
LoRA	43mins	1h37mins	4h12mins	2h29mins	1h56mins	1h22mins
Full Fine Tuning	53mins	1h42mins	2h57mins	1h59mins	1h59mins	2h2mins

(c) The training time for each experiment

Table 3. LoRA and full fine-tuning details for Frisian and English.

3.4.3 Full Fine-Tuning Experiments

Similarly, six full fine-tuning experiments were conducted for the study. All Frisian models were trained using learning rate 1e-5, training batch size 8, evaluation batch size 8, 50 steps warm-up, 1 gradient accumulation steps, Adam optimizer with 1e-08 epsilon, seed 42, and steps as evaluation strategy. The 10-minute model was trained for 250 steps, the 1-hour model for 1000 steps, and the 10-hour model for 1500 steps. Same as LoRA Frisian experiments, the language in the whisper tokenizer and whisper processor were also all set to Dutch.

All English models were trained using learning rate 5e-7, training batch size 8, evaluation batch size 8, 300 steps warm-up, 2 gradient accumulation steps, Adam optimizer with 1e-08 epsilon, seed 42, and steps as evaluation strategy. In addition, dropout 0.4 and early stopping patience 5 were applied to the model.

3.5 Ethical Consideration

When it comes to the ethical concerns in ASR research, data privacy and replicability are addressed in this study. The human voice is considered as important biometric data that can be used to identify individuals, therefore the voice data should be carefully treated and should not be traced back to any individuals. The Common Voice Corpus and LibriSpeech datasets used in the current study are all publicly available. The recordings from the former corpus are collected in a crowdsourced way and only demographic metadata such as gender, age, and accents for each sentence is reported, so no individual can be identified. The latter corpus is freely available under the CC BY 4.0 license[6]. The study also does not require any subjective evaluations from participants, so consents are not necessary.

The other ethical considerations regarding replicability will be resolved by sharing codes and models to the public. The code will be available at the GitHub repository[7], and the model checkpoints will be shared in HuggingFace[8]. The experiments can be reproduced with the shared code, but the results might differ slightly due to some randomness in the model.

[6] https://creativecommons.org/
[7] https://github.com/xuliu15/VT_Thesis
[8] https://huggingface.co/collections/xuliu15/vt-thesis-models-666845144eb7ca960ebc88a4

4 Results and Discussion

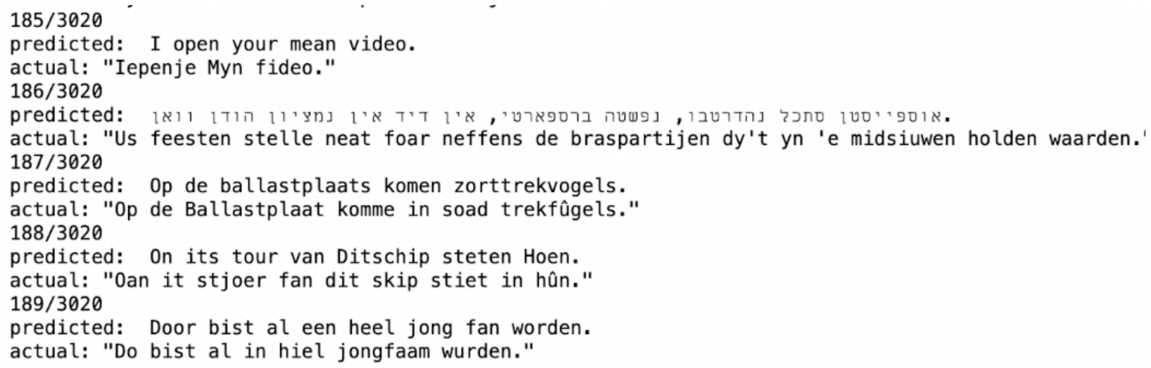
In this chapter, I will display and analyze the results of the experiments based on my hypothesis regarding the WER, number of trainable parameters, and GPU memory usage. I will also discuss and compare my findings with previous literature.

4.1 Word Error Rate

The results regarding WER for zero-shot, LoRA, and full fine-tuning experiments can be found in Table 4 and Figure 6. From Table 4 it can be seen that without any training, the WER of Frisian on Zero-shot Whisper-small was 87.89%, in other words, only less than 13% of Frisian sentences could be correctly recognized. Figure 5 shows part of the transcriptions. As seen in this figure, the predicted transcriptions were very poor, in which some of the predicted sentences were in completely different languages, not even close to Dutch. The undesired performance indicates the need for improving Whisper to better recognize the Frisian language. Zero-shot English performed very well on Whisper small with 3.9% WER. This result is even better than the officially reported 6.7% WER for Whisper small on LibriSpeech test-clean.

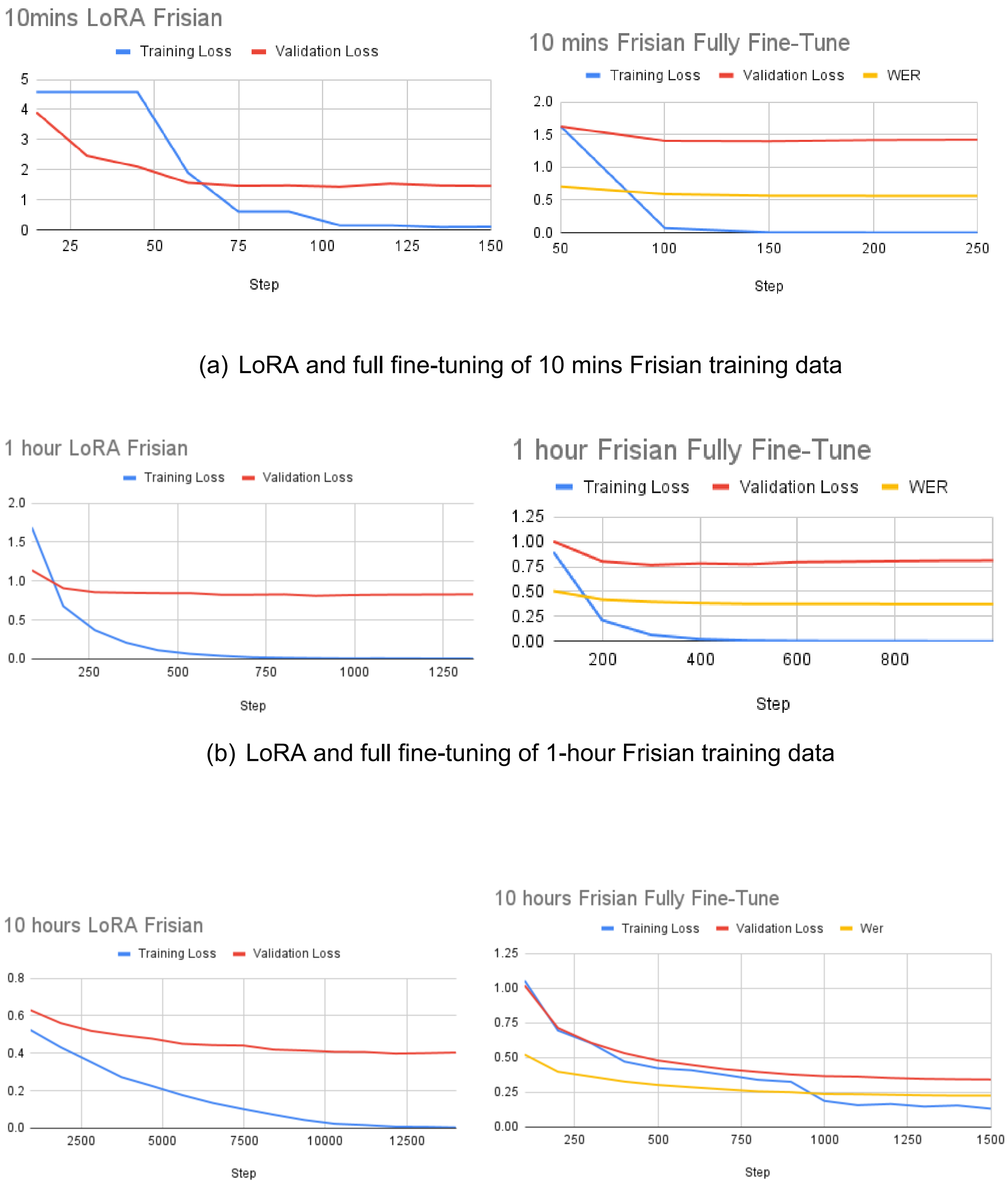
WER	Frisian		English	
Model	Full Fine Tuning	LoRA	Full Fine Tuning	LoRA
Zero-Shot	87.89		3.9	
10mins	**56.25**	59.55	**3.41**	5.33
1h	**37.58**	39.6	**3.45**	4.96
10h	**22.43**	23.5	**3.45**	4.11

Table 4. The WER for Frisian and English in all experiments.



```
185/3020
predicted:  I open your mean video.
actual: "Iepenje Myn fideo."
186/3020
predicted:  אוספייסטן סתכל נהדרטבו, נפשטה ברספארטי, אין דיד אין נמציון הודן וואן.
actual: "Us feesten stelle neat foar neffens de braspartijen dy't yn 'e midsiuwen holden waarden.'
187/3020
predicted:  Op de ballastplaats komen zorttrekvogels.
actual: "Op de Ballastplaat komme in soad trekfûgels."
188/3020
predicted:  On its tour van Ditschip steten Hoen.
actual: "Oan it stjoer fan dit skip stiet in hûn."
189/3020
predicted:  Door bist al een heel jong fan worden.
actual: "Do bist al in hiel jongfaam wurden."
```

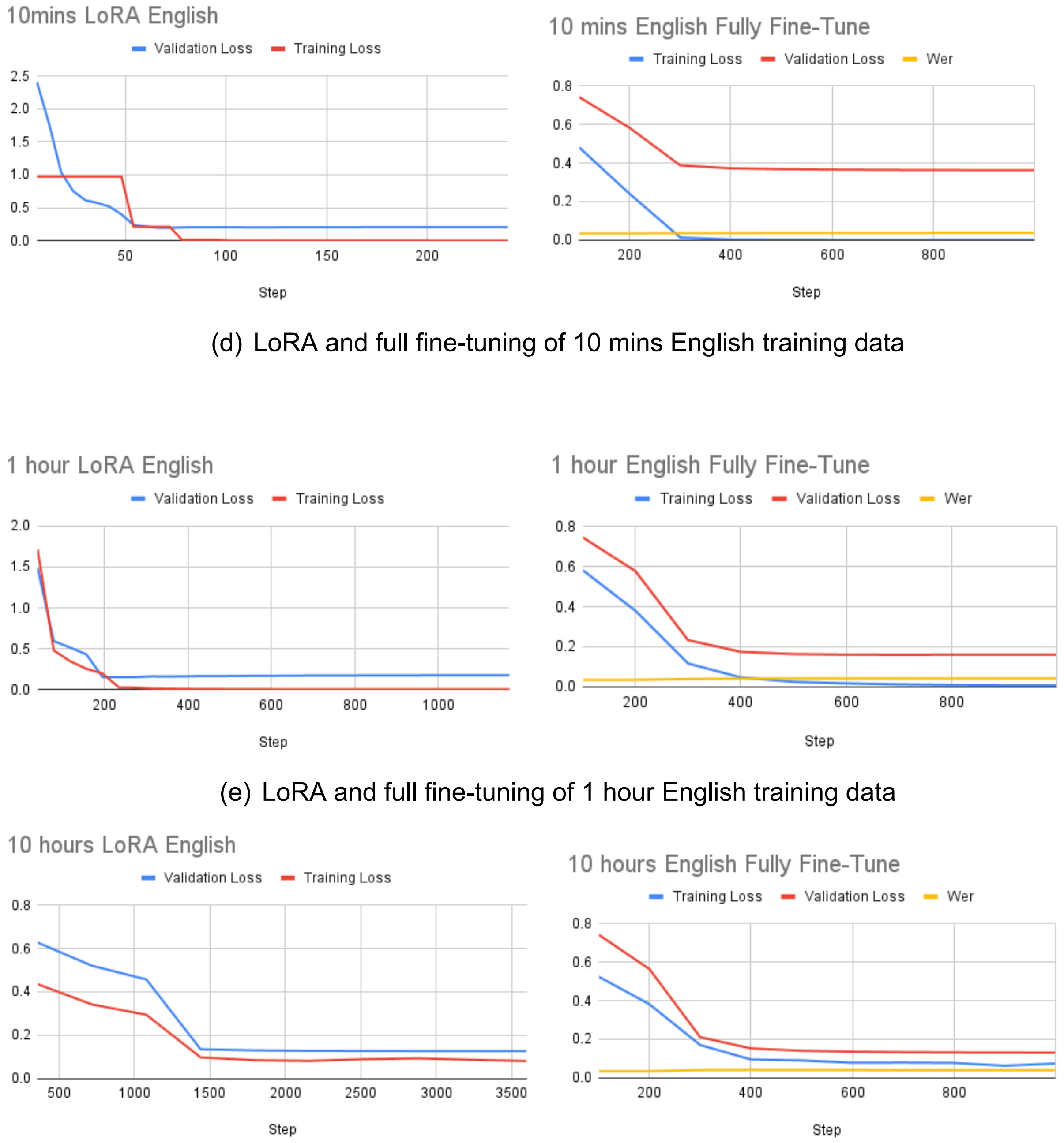
Figure 5. The zero shot evaluation results for Frisian on Whisper-small.



(a) LoRA and full fine-tuning of 10 mins Frisian training data

(b) LoRA and full fine-tuning of 1-hour Frisian training data

(c) LoRA and full fine-tuning of 10 hours Frisian training data



(d) LoRA and full fine-tuning of 10 mins English training data

(e) LoRA and full fine-tuning of 1 hour English training data

(f) LoRA and full fine-tuning of 10 hours English training data

Figure 6. The training loss, validation loss, and WER of LoRA and full fine-tuning for 10 mins, 1-hour, and 10 hours of Frisian training data.

In terms of LoRA and full fine-tuning experiments for Frisian, it can be seen that with only 10 mins of training data, LoRA greatly reduced the WER from 87.89% to 59.55%, and full fine-tuning was about 3% lower than LoRA. 1 hour of training data further reduced the WER of LoRA by 20% to 39.6%, and the WER of full fine-tuning also decreased to 37.58%. Trained with 10 hours of speech data, LoRA achieved 23.5% WER while full fine-tuning had slightly better WER at 22.43%. Overall, as the size of training data increased, the WER reduced for both LoRA and fine-tuning. LoRA was able to achieve comparable WER as full fine-tuning at all data sizes, despite slightly worse results. However, this performance gap gradually decreased as more training data were involved. It might be possible that with even more training data of 20 hours or so, LoRA could outperform full fine-tuning.

As for English, it can be observed that compared with the zero-shot model, fully fine-tuning with 10 minutes, 1 hour, and 10 hours of training data all improved the model performance by around 0.5% WER. However, the WER of full fine-tuning models did not decrease as the size of training data increased. Instead, they stayed more or less the same around 3.4%. All LoRA models, on the other hand, underperformed the zero-shot model and full fine-tuning models. With 10 mins of training data, LoRA achieved 5.33% WER and it reduced to 4.96% with 1 hour of training data. WER further reduced to 4.11% with 10 hours of training data. All in all, full fine-tuning models outperforms LoRA at all data sizes with slightly better performance for English.

From Figure 6, it can be seen that all models have reached convergence. For full fine-tuning, the gap between validation loss and training loss was narrowing down as the size of training data grew. Compared with LoRA at different data sizes, full fine-tuning always had a wider gap than LoRA, except for 10 hours Frisian. This suggests that fine-tuning might be prone to overfit when data size was small.

4.2 Number of Trainable Parameters

In the experiments, LoRA were inserted into value and query projection matrices in the self-attention modules. By setting $r = 32$, $\alpha = 64$, dropout = 0.05 and bias = none, the size of LoRA was 3,538,944. As seen in Table 6, although the insertion of LoRA increased the total parameters of the model, the parameters that needed to be trained only took 1.4% of all parameters. On the contrary, 99.5% of all parameters needed to be trained for full fine-tuning.

Methods	All parameters	Trainable Parameters	Percent
Full Fine-Tuning	241,734,912	240,582,912	99.5%
LoRA	245,273,856	3,538,944	1.4 %

Table 6. The number of parameters for two methods.

4.3 GPU Memory Usage

For Frisian, using full fine-tuning methods required a larger amount of GPU memory than LoRA. As seen from Table 7, LoRA consumed between 6400MiB to 6800MiB whereas full fine-tuning needed more than 8800 MiB in the training process. For English, full fine-tuning used less GPU memory than LoRA. This is because dropout 0.4 was applied in English full fine-tuning training configuration, but not in LoRA configuration. I tested on disabling dropout in full fine-tuning and the GPU memory went up to over 10000MiB.

GPU Memory	Frisian		English	
	LoRA	Full Fine-Tuning	LoRA	Full Fine-Tuning
10mins	6406MiB	9080 MiB	7504MiB	6457MiB
1h	6404 MiB	8804 MiB	7780MiB	6142MiB
10h	6748 MiB	8804MiB	7784MiB	6395MiB

Table 7. The GPU memory usage during the training of LoRA and fine-tuning.

4.4 Discussion

It is evident that LoRA is very effective in tuning the pre-trained multilingual ASR Whisper model to recognize the low-resource language Frisian. My hypothesis that LoRA is able to achieve significant WER reduction for Frisian which will be the comparable performance to full fine-tuning is validated. The other hypothesis regarding less trainable parameters and less GPU memory usage in LoRA than in full fine-tuning has also been supported.

First, with only 10 mins of Frisian training data, LoRA is capable of reducing the zero-shot WER by 28%. Increasing the size of training data to 1 hour and 10 hours further improved the LoRA performance by 20% and 16% respectively. Despite the LoRA achieving slightly worse WER than fully fine-tuning, the difference is really small between 1% to 3%. Further increasing the training data size has the potential to reduce the performance gap between LoRA and full fine-tuning. The reasonable performance of LoRA for Frisian is in line with previous studies of LoRA on low-resource languages such as Malayalam and Galician where they showed LoRA reduced the WER compared with zero-shot and achieved similar results as in full fine-tuning (Ferraz et al., 2024b; Kim et al., 2023).

Second, the current Frisian WER results of LoRA are achieved with only 1.4% of all parameters of the Whisper small model, and with only ⅔ to ¾ of GPU memory that is needed for fine-tuning. It shows that LoRA does not require too many computational resources as full fine-tuning does and is therefore very cost-efficient. The finding about parameters is similar to other LoRA studies where they also reported that LoRA successfully tuned the model for low-resource languages with a small fraction of parameters (Ferraz et al., 2024b; Kim et al., 2023).

It is, however, interesting to see the performance of LoRA and full fine-tuning for English is a bit unexpected. All three LoRA models underperformed zero-shot and its corresponding full fine-tuning model. The WER of full fine-tuning models also did not decrease with larger training data size. The degraded performance of LoRA in English is aligned with the finding from (Z.-C. Chen et al., 2023). In their study, they trained self-supervised speech models with 1 hour and 10 hours of Libri-Light dataset and tested the LoRA performance on the testing dataset of LibriSpeech. They found that LoRA failed to achieve a comparable performance for English low-resource scenarios because LoRA performed poorly with higher WER than baseline models and full fine-tuning models. Until the size of training data reached 100 hours, it can finally be seen that LoRA achieved close WER as full fine-tuning, baseline, and other PEFT methods.

Because of the poor performance of LoRA in English, (Z.-C. Chen et al., 2023) claimed that LoRA in general cannot perform well in speech tasks, which I disagree with. The current study strongly shows that LoRA has great power when it comes to adapting models to low-resource languages. It just could be the case that low-resource languages that are unseen or that take only a tiny proportion in the pre-trained model can benefit more from LoRA than those high-resource languages. For high-resource languages such as English, it takes a larger size of training data to reveal LoRA’s full potential. This can be seen from (Z.-C. Chen et al., 2023)’s study, as the size of training data increased to 100 hours, LoRA started to reach comparable results as in full fine-tuning. Another reason for the unexpected performance of LoRA and full fine-tuning for English could be that the zero-shot Whisper small is already good enough for the LibriSpeech dataset, so there is no room for further improvement (Do et al., 2023).

5 Conclusion

This study aims to investigate the effectiveness of one PEFT method, namely LoRA, in adapting the large pre-trained multilingual ASR model to recognize low-resource language Frisian. The efficiency is evaluated via several metrics including the WER, number of trainable parameters, and GPU memory usage. The Whisper small model is either fully fine-tuned or tuned with LoRA using various sizes of Frisian and English training data, ranging from 10 minutes, 1 hour, to 10 hours. The results are compared between the full fine-tuning method and the LoRA method.

Results showed that using only 1.4% of all model parameters and less GPU memory, LoRA is able to greatly reduce Frisian WER. Compared with zero-shot evaluation, Frisian WER has been reduced by 28%, 48%, and 64% with 10 minutes, 1 hour, and 10 hours of training data, respectively. The Frisian WER achieved by LoRA is very comparable to fully fine-tuning, with only a 1% to 3% gap. The results also showed that low-resource languages such as Frisian benefit more from LoRA than high-resource languages such as English.

In conclusion, this study timely addresses a practical challenge in speech technology by assessing the applicability of PEFT, particularly LoRA, on multilingual ASR models in low-resource contexts. It deepens the theoretical understanding of state-of-the-art model tuning techniques and brings valuable insights for practical ASR system development toward efficiency and inclusion.

5.1 Limitations

Speaking of limitations of the current research, it can be concluded that the scope of the study is rather narrowed because the study only investigated the Whisper small model and the small size of the training datasets from two languages. Therefore, it might not be suitable to generalize the results to a broader scope, for example, to all low-resource languages or all ASR models. In the future, the study could be expanded to a larger model such as the Whisper-large model and a larger training dataset for example 50 hours or 100 hours of training data if available for low-resource languages. In the case of Frisian, one can take advantage of the newest Common Voice Corpus 16.1 which contains 69 hours of validated speech. In this way the efficiency of LoRA can be even further investigated, especially exploring if the LoRA would outperform full fine-tuning when more training data are used. One can also explore if with LoRA and a large size of training data, the Whisper model is able to achieve state-of-the-art WER.

Another limitation of the current study is that it only focuses on the insertion of LoRA modules into value and query projection matrices in the self-attention module. As said, LoRA could be applied on different attention weights matrices such as query, key, value, and output matrices, or different layers such as MLP layers, LayerNorm layers, and biases (Hu et al., 2021). In (Z.-C. Chen et al., 2023)'s study, compared with other adapters that were added behind the second feed-forward layer, LoRA that was added in the self-attention modules performed the worst in the SUPERB benchmark. The authors suggested that the position the adapter added resulted in such performance discrepancy. Therefore, it might be worth exploring in the future how different positions affect the performance of LoRA for low-resource languages.

5.2 Future Research

In the future, it might be interesting to work on different PEFT methods to find more flexible and efficient ways for training and deploying models for low-resource languages. One way is to combine two different techniques to further reduce computational demand and improve performance. For example, in (Kim et al., 2023)' study, the authors combined the pruning strategy Lottery Ticket Hypothesis with LoRA on the Whisper model for low-resource languages, and the model achieved better performance than the LoRA-only approach. The other possibilities could be the combination between quantization and LoRA, or the insertion of LoRA on distilled pre-trained models. The second way is to compare the efficiency of several PEFT methods under low-resource language scenarios and explore the most efficient configuration of that PEFT method.

In addition, it would be valuable to discover how to preserve the multilingual capability of Whisper while adapting to LoRA modules. In (Ferraz et al., 2024b)'s study, they proposed a DistilWhisper model that retained the multilingual and multitasking performance of the original Whisper model. They preserve the multilingual capability by training conditional language-specific routing (CLSR) modules with gated mechanisms in parallel and loading the relevant modules at the inference. Inspired by this approach, it might also be possible to couple and train several language-specific LoRA modules and load individual ones when necessary.

References

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., & Weber, G. (2019). Common voice: A massively-multilingual speech corpus. *arXiv Preprint arXiv:1912.06670*.

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. (2021). *XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale* (arXiv:2111.09296). arXiv. http://arxiv.org/abs/2111.09296

Bapna, A., Cherry, C., Zhang, Y., Jia, Y., Johnson, M., Cheng, Y., Khanuja, S., Riesa, J., & Conneau, A. (2022). *mSLAM: Massively multilingual joint pre-training for speech and text* (arXiv:2202.01374). arXiv. http://arxiv.org/abs/2202.01374

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. http://arxiv.org/abs/2005.14165

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., & Wei, F. (2022). WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1505–1518. https://doi.org/10.1109/JSTSP.2022.3188113

Chen, Z., Zhang, Y., Rosenberg, A., Ramabhadran, B., Moreno, P., Bapna, A., & Zen, H. (2022). *MAESTRO: Matched Speech Text Representations through Modality Matching* (arXiv:2204.03409). arXiv. http://arxiv.org/abs/2204.03409

Chen, Z.-C., Fu, C.-L., Liu, C.-Y., Li, S.-W., & Lee, H. (2023). *Exploring Efficient-tuning Methods* *in Self-supervised Speech Models* (arXiv:2210.06175). arXiv. http://arxiv.org/abs/2210.06175

Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., & Bapna, A. (2022). *FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech* (arXiv:2205.12446). arXiv. http://arxiv.org/abs/2205.12446

Do, A., Brown, O., Wang, Z., Mathew, N., Liu, Z., Ahmed, J., & Yu, C. (2023). Using fine-tuning and min lookahead beam search to improve Whisper. *arXiv Preprint arXiv:2309.10299*.

Fathullah, Y., Wu, C., Lakomkin, E., Jia, J., Shangguan, Y., Li, K., Guo, J., Xiong, W., Mahadeokar, J., Kalinli, O., Fuegen, C., & Seltzer, M. (2023). *Prompting Large Language Models with Speech Recognition Abilities* (arXiv:2307.11795). arXiv. http://arxiv.org/abs/2307.11795

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. *2023 11th International Conference on Affective Computing and Intelligent Interaction (ACII)*, 1–8. https://doi.org/10.1109/ACII59096.2023.10388152

Ferraz, T. P., Boito, M. Z., Brun, C., & Nikoulina, V. (2024b). 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)*.

He, C.-Y., & Chien, J.-T. (2023). Learning Adapters for Code-Switching Speech Recognition. *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 344–349. https://doi.org/10.1109/APSIPAASC58517.2023.10317410

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2021). Towards a unified view of parameter-efficient transfer learning. *arXiv Preprint arXiv:2110.04366*.

He, S., Ding, L., Dong, D., Zhang, M., & Tao, D. (2022). *SparseAdapter: An Easy Approach for*

Improving the Parameter-Efficiency of Adapters (arXiv:2210.04284). arXiv. http://arxiv.org/abs/2210.04284

Hou, W., Dong, Y., Zhuang, B., Yang, L., Shi, J., & Shinozaki, T. (2020). Large-scale end-to-end multilingual speech recognition and language identification with multi-task learning. *Babel*, *37*(4k), 10k.

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.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models* (arXiv:2106.09685). arXiv. http://arxiv.org/abs/2106.09685

Javed, T., Doddapaneni, S., Raman, A., Bhogale, K. S., Ramesh, G., Kunchukuttan, A., Kumar, P., & Khapra, M. M. (2022). *Towards building asr systems for the next billion users*. *36*(10), 10813–10821.

Kannan, A., Datta, A., Sainath, T. N., Weinstein, E., Ramabhadran, B., Wu, Y., Bapna, A., Chen, Z., & Lee, S. (2019). *Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model* (arXiv:1909.05330). arXiv. http://arxiv.org/abs/1909.05330

Kim, H. S., Cho, C. H., Won, H., & Park, K. H. (2023). Adapt and Prune Strategy for Multilingual Speech Foundational Model on Low-resourced Languages. *Proceedings of the 3rd Workshop on Multi-Lingual Representation Learning (MRL)*, 85–94.

Le, H., Pino, J., Wang, C., Gu, J., Schwab, D., & Besacier, L. (2021). *Lightweight Adapter Tuning for Multilingual Speech Translation* (arXiv:2106.01463). arXiv. http://arxiv.org/abs/2106.01463

Lester, B., Al-Rfou, R., & Constant, N. (2021). *The Power of Scale for Parameter-Efficient Prompt Tuning* (arXiv:2104.08691). arXiv. http://arxiv.org/abs/2104.08691

Li, B., Pang, R., Sainath, T. N., Gulati, A., Zhang, Y., Qin, J., Haghani, P., Huang, W. R., Ma, M.,

& Bai, J. (2021). *Scaling End-to-End Models for Large-Scale Multilingual ASR* (arXiv:2104.14830). arXiv. http://arxiv.org/abs/2104.14830

Li, X. L., & Liang, P. (2021). *Prefix-Tuning: Optimizing Continuous Prompts for Generation* (arXiv:2101.00190). arXiv. http://arxiv.org/abs/2101.00190

Liu, W., Qin, Y., Peng, Z., & Lee, T. (2024). *Sparsely Shared LoRA on Whisper for Child Speech Recognition* (arXiv:2309.11756). arXiv. http://arxiv.org/abs/2309.11756

Mena, C., Gatt, A., DeMarco, A., Borg, C., Van der Plas, L., Muscat, A., & Padovani, I. (2020). Masri-headset: A maltese corpus for speech recognition. *arXiv Preprint arXiv:2008.05760*.

Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: A large-scale speaker identification dataset. *arXiv Preprint arXiv:1706.08612*.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An asr corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210.

Peng, J., Stafylakis, T., Gu, R., Plchot, O., Mošner, L., Burget, L., & Černocký, J. (2022). *Parameter-efficient transfer learning of pre-trained Transformer models for speaker verification using adapters* (arXiv:2210.16032). arXiv. http://arxiv.org/abs/2210.16032

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., & Gurevych, I. (2020). Adapterfusion: Non-destructive task composition for transfer learning. *arXiv Preprint arXiv:2005.00247*.

Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020). *MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer* (arXiv:2005.00052). arXiv. http://arxiv.org/abs/2005.00052

Pratap, V., Sriram, A., Tomasello, P., Hannun, A., Liptchinsky, V., Synnaeve, G., & Collobert, R. (2020). Massively multilingual ASR: 50 languages, 1 model, 1 billion parameters. *arXiv Preprint arXiv:2007.03001*.

Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., Elkahky, A., Ni, Z., Vyas, A., Fazel-Zarandi, M., Baevski, A., Adi, Y., Zhang, X., Hsu, W.-N., Conneau, A., & Auli, M. (2023). *Scaling Speech Technology to 1,000+ Languages* (arXiv:2305.13516). arXiv. http://arxiv.org/abs/2305.13516

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., & Collobert, R. (2020). MLS: A Large-Scale Multilingual Dataset for Speech Research. *Interspeech 2020*, 2757–2761. https://doi.org/10.21437/Interspeech.2020-2826

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022a). *Robust Speech Recognition via Large-Scale Weak Supervision*. https://doi.org/10.48550/ARXIV.2212.04356

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*.

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. *INTERSPEECH 2023*, 1958–1962. https://doi.org/10.21437/Interspeech.2023-1407

Rebuffi, S.-A., Bilen, H., & Vedaldi, A. (2017). *Learning multiple visual domains with residual adapters* (arXiv:1705.08045). arXiv. http://arxiv.org/abs/1705.08045

Rouditchenko, A., Khurana, S., Thomas, S., Feris, R., Karlinsky, L., Kuehne, H., Harwath, D., Kingsbury, B., & Glass, J. (2023). *Comparison of Multilingual Self-Supervised and Weakly-Supervised Speech Pre-Training for Adaptation to Unseen Languages* (arXiv:2305.12606). arXiv. http://arxiv.org/abs/2305.12606

Southwell, R., Ward, W., Trinh, V. A., Clevenger, C., Clevenger, C., Watts, E., Reitman, J., D’Mello, S., & Whitehill, J. (2024). Automatic Speech Recognition Tuned for Child Speech in the Classroom. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12291–12295.

Tomanek, K., Zayats, V., Padfield, D., Vaillancourt, K., & Biadsy, F. (2021). *Residual Adapters for*

Parameter-Efficient ASR Adaptation to Atypical and Accented Speech (arXiv:2109.06952). arXiv. http://arxiv.org/abs/2109.06952

Wang, C., Rivière, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., & Dupoux, E. (2021). *VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation* (arXiv:2101.00390). arXiv. http://arxiv.org/abs/2101.00390

Williams, A., Demarco, A., & Borg, C. (2023). *The Applicability of Wav2Vec2 and Whisper for Low-Resource Maltese ASR*. 39–43.

Xu, L., Xie, H., Qin, S.-Z. J., Tao, X., & Wang, F. L. (2023). *Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment* (arXiv:2312.12148). arXiv. http://arxiv.org/abs/2312.12148

Yi, C., Wang, J., Cheng, N., Zhou, S., & Xu, B. (2020). Applying wav2vec2. 0 to speech recognition in various low-resource languages. *arXiv Preprint arXiv:2012.12121*.

Yilmaz, E., van den Heuvel, H., Dijkstra, J., Van de Velde, H., Kampstra, F., Algra, J., & Van Leeuwen, D. (2016). Open source speech and language resources for Frisian. *Interspeech 2016*, 1536–1540.

Yılmaz, E., van den Heuvel, H., & Van Leeuwen, D. (2016). Investigating bilingual deep neural networks for automatic recognition of code-switching frisian speech. *Procedia Computer Science*, *81*, 159–166.

Zaken, E. B., Ravfogel, S., & Goldberg, Y. (2021). Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv Preprint arXiv:2106.10199*.

Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., Chen, Z., Chen, N., Li, B., Axelrod, V., Wang, G., Meng, Z., Hu, K., Rosenberg, A., Prabhavalkar, R., Park, D. S., Haghani, P., Riesa, J., Perng, G., Soltau, H., … Wu, Y. (2023, March 2). *Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages*. arXiv.Org. https://arxiv.org/abs/2303.01037v3

Zhao, J., & Zhang, W.-Q. (2022). Improving automatic speech recognition performance for low-resource languages with self-supervised models. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1227–1241.