Fine-tuning OpenAI's Whisper for Multilingual ASR with Transformers

Sophia Kang, Phuc Duong

Department of Computer Science

Yale University

New Haven, CT 06520

sophia.kang@yale.edu, phuc.duong@yale.edu

Abstract

In this project, we investigate and build on Whisper, OpenAI's automatic speech recognition (ASR) model, as detailed in the paper "Robust Speech Recognition via Large-Scale Weak Supervision," to focus on accurate recognition and transcription of Vietnamese and Korean. These languages present unique linguistic challenges in the speech recognition space: Vietnamese, with its tonal marks, and Korean, with word segmentation. Through Mozilla's Common Voice and Google's FLEURS datasets and leveraging OpenAI's Whisper API and Hugging Face's transformers library to train and load the model, we test whether fine-tuning Whisper on additional datasets reduces transcription errors and improves adaptability in regards to the model ability to handle the complexity of these two languages. Our code is available at https://github.com/phucd5/whisper-asr.

1 Project Motivation

Speech recognition technology has become increasingly important in a variety of applications, ranging from virtual assistants to automated transcription services. There is a growing demand for models that can accurately recognize and transcribe speech in multiple languages, but not all languages are high-resource. The recently released (September 2022) Whisper model by OpenAI has proved to be effective in capabilities in speech recognition. In this project, we plan to refine this model by enhancing its performance in multilingual contexts. This project will focus on fine-tuning Whisper to better handle multilingual inputs, thereby aiming to contribute to a more inclusive and effective speech recognition technology. We also hope to gain practice fine-tuning a pre-trained model by leveraging multiple datasets.

2 Related Work

The Whisper model is presented in "Robust Speech Recognition via Large-Scale Weak Supervision." This model is scaled to 680,000 hours of multilingual and multitask (transcription and translation), and the authors present the model as performing competitive pre-trained results without the need to fine-tune. Although speech recognition is mostly evaluated on Word Error Rate (WER), the authors standardize the text extensively before the calculation of WER to avoid calculating based on string edit distance. Datasets used for analysis include Artie, Common Voice, FLEURS En, Tedlium, CHiME6, VoxPopuli En, CORAAL, AMI IHM, Switchboard, CallHome, WSJ, AMI SDM1, and LibriSpeech Other.

Preprint. Under review.

Meanwhile, making technologies such as ASR robust for low resource languages has been the subject of recent research. Singh et al. (2023) addressed the issue of having scarce annotated data hindering development of accurate ASR systems by proposing a self-training approach that generates pseudo-labels for unlabeled low-resource speech. Liu et al. (2023) examined the impact of language model (LM) size on low-resource ASR and found that having larger LMs do not necessarily result in lower WER. In particular, larger LMs resulted in significantly worse performance in the most low resource language out of the five widely-spoken low resource languages they studied. Making models robust for low resource languages, however, is not only a problem for the ASR domain and is also relevant for many other NLP tasks, such as Neural Machine Translation (NMT). For instance, Ranathunga et al. (2023) observe that multi-NMT models trained with roughly 50 languages show clear performance gains over bilingual models for low resource languages, as the model learns an "*interlingua*," or shared semantic representation between languages.

This paper builds on previous work in that it tests if fine-tuning does improve the Whisper model's performance in its ability to transcribe multilingual audio inputs. We also hope to contribute to the conversations on low resource languages by examining how a large industry model such as Whisper performs when it is fine-tuned on publicly available datasets for low resource languages. We also test if the model is able to develop better interlingua as a result of fine-tuning with different datasets.

3 Proposed Approach

3.1 Fine-tuning Model

We use OpenAI's pre-trained whisper-small model available through the Hugging Face transformers library via WhisperProcessor and specify the task as transcription. We chose the whisper-small model because of compute constraints. Whisper is of an encoder-decoder Transformer structure, where the encoder input is audio split into 30 second chunks and converted into log-Mel spectrograms. The decoder is trained to predict the corresponding text transcription to audio input, and is passed special tokens that enable the single model to perform multiple tasks such as multilingual speech transcription and to-English speech translation.

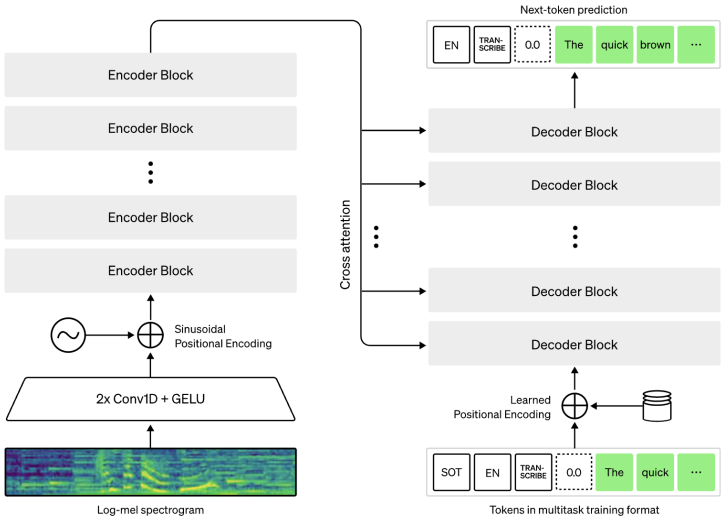

Figure 1: Whisper Model Architecture Image Credit: OpenAI.

3.2 Dataset Choice

We load both datasets, Mozilla Common Voice 13.0 and Google FLEURS, using the datasets library, and fine-tune the model on these datasets. The Mozilla Common Voice dataset consists of a MP3 and

corresponding reference transcript file, and contains 19,160 hours of data in 114 languages. However, the distribution of hours across languages is not uniform. On the other hand, the FLEURS dataset includes approximately 12 hours of speech supervision for each of the 102 languages it includes. We leverage both datasets because we discovered that while the Common Voice dataset contains comparatively ample data for Vietnamese (2.46k rows of training data), it does not for Korean (192 rows of training data). For each dataset, we make sure that the input audio files have a sampling rate of 16kHz, which is the sampling rate expected by the Whisper model. The Mozilla Common Voice and Google FLEURS datasets can be found at the following sites:

`https://huggingface.co/datasets/mozilla-foundation/common_voice_13_0`
`https://huggingface.co/datasets/google/fleurs`

3.3 Model Setup and Training

Our model was trained on the following parameters:

Parameter	Value
Train Batch Size	16
Gradient Accumulation Steps	1
Learning Rate	1e-5
Warm Up Steps	500
Max Steps	4000
Eval Batch Size	8

Table 1: Hyperparameters for model training

These parameters were chosen based on Hugging Face's attempt to fine-tune Whisper on the Hindi language. It can also be noted that Jong Wook Kim, an author of the Whisper paper, suggested a fine-tuning learning rate that is 40x smaller than what was used for pre-training, which roughly aligns with the parameters we used.

We used an NVIDIA RTX A5000 graphics card for training and evaluation on the Yale High Performance Computing (HPC) clusters, running the Red Hat Enterprise Linux OS with version 8.8, codenamed Ootpa. We used an Intel Xeon Gold 6326 CPU with 64 CPU cores, with 64 GB of RAM available on the cluster, and 4 CPUs allocated per task.

3.4 Industry Models Tested

Industry models were tested for comparison with our fine-tuned models. We tested speech-to-text models from Google and IBM by using their respective APIs and then training on the Common Voice dataset. While we initially proposed that we would test the Microsoft Azure speech-to-text API, we were unable to set up the speech configs due to its incompatibility with the mp3 files in the Common Voice dataset. As a result, we instead proceeded with evaluating the models from Google and IBM. In addition, we chose only to use the Common Voice dataset for evaluation in this section due to the technical constraints that came with the .wav file in FLEURS dataset and its interactions with the APIs.

3.5 Evaluation Metrics

We evaluated the models using Word Error Rate (WER), the standard metric for speech recognition evaluation, and then employ two more metrics, each one designated for each language. For Korean, we measure the spacing error rate, to measure words are segmented correctly. Because of difficulty using existing packages such as PyKoSpacing locally, we instead implemented our own spacing function that examines each reference and prediction pair. The accuracy rate is calculated as $1 - E/N$, where E stands for total number of spacing errors in reference - prediction pairs and N stands for total number of characters in reference strings. This metric design was in part inspired by the WER, which

is calculated by $\frac{S+D+I}{N}$ where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the number of words in reference, or equivalently $S + D + C$, where C is the number of correct words. To account for minor differences in transcription, such as puncutation marks, all puncutation marks were removed and then compared for spacing. For Vietnamese, we use Character Error Rate (CER), available through the evaluate library, to measure whether tones are marked correctly. This was calculated similarly to WER but deals with characters, and is computed as $\frac{S+D+I}{N}$.

4 Results

The following tables summarize our findings for each model and for each language. CV stands for the model fine-tuned on the Common Voice dataset only, CV+FLEURS stands for the model fine-tuned on the Common Voice and FLEURS datasets. There are two values for each error rate (WER, CER, and Spacing accuracy), because we evaluate on test sets from CV and FLEURS, respectively.

	Pre-trained	CV	CV+FLEURS
WER	CV: 40.4628, FLEURS: 28.9132	CV: 27.2176, FLEURS: 42.6675	CV: 40.1322, FLEURS: 28.5836
CER	CV: 18.7665, FLEURS: 13.005	CV: 11.8753, FLEURS: 20.4371	CV: 17.188, FLEURS: 13.2312
Spacing	CV: 95.7531, FLEURS: 93.9761	CV: 96.4823, FLEURS: 92.7131	CV: 95.8573, FLEURS: 95.3187

Table 2: Results for Vietnamese

4.1 Vietnamese Results

Initially, the pre-trained model demonstrated a 40.4628% WER on the Common Voice (CV) dataset and a 28.9132% WER on Google's FLEURS dataset. After fine-tuning it on the CV dataset, the WER decreased to 27.2176%, reflecting roughly a 13.25% improvement in word accuracy. However, evaluation on the FLEURS dataset led to worse performance than our original pre-trained model, with an increase of 2.20% to a 42.6675% WER. This points to the fact that the model has become over-specialized in linguistic patterns in the CV dataset, undermining its generalizability in the FLEURS dataset.

Subsequent fine-tuning on both the FLEURS and CV datasets found more promising results, as it helped mitigate this overfitting. Performance on the CV dataset increased to 40.1322%, which, while worse than the CV-only fine-tuned model, still represents an improvement over the original pre-trained model. On the other hand, the FLEURS dataset resulted in a 28.5836%, surpassing the performance of both the CV-only fine-tuned model and the retrained model. These results indicate that training both on both datasets, CV, and FLEURS, resulted in an increased performance from our base model with a slight decrease of 0.33% and 0.38% for the CV and Fleurs datasets, respectively, showing that the model did benefit from more diverse datasets when fine-tuning with respect to the WER and led to the model improved ability to transcribed Vietnamese speech to text.

CER follows a similar trend to WER, with the pre-training model having a CER of 18.7665% on CV and 13.005% on FLEURS. After fine-tuning with the CV dataset, the fine-tuned model resulted in a decrease in CER to 11.8753% for the CV dataset but resulted in a 7.4321% CER increase to a 20.4371% CER. Similar to WER, our model shows overfitting to the CV dataset. Fine-tuning the CV + FLEURS dataset reversed the overfitting. The model fine-tuned on both datasets resulted in a 1.5785% decrease on the CV dataset with a 17.188% WER but led to a 0.2262% increase with a 13.2312% CER on the FLEURS dataset. This new model gives a more balanced approach, reducing the error rate for CV but does not increase the error rate for FLEURS as much as it did when we only fine-tuned on CV, maintaining an overall solid performance on both datasets.

While fine-tuning does show an improved understanding of tonal characters, when looking at the CV dataset, fine-tuning on specific datasets can lead to overfitting, as indicated by the marginal improvement and the decrease in performance depending on the dataset we evaluated.

	Pre-trained	CV	CV+FLEURS
WER	CV: 31.4199, FLEURS: 23.5993	CV: 32.1249, FLEURS: 33.5461	CV: 33.8369, FLEURS: 28.7083
CER	CV: 10.4123, FLEURS: 9.0974	CV: 10.595, FLEURS: 13.51	CV: 10.8559, FLEURS: 10.2842
Spacing	CV: 97.286, FLEURS: 96.9985	CV: 97.0251, FLEURS: 96.0733	CV: 97.2599, FLEURS: 97.211

Table 3: Results for Korean

4.2 Korean Results

Evaluations on the CV test set show that WER increases as we fine-tune it on more data values from 31.4199% for the pre-trained Whisper model to 32.1249% for the model fine-tuned on CV and 33.8369% for the model fine-tuned on CV and FLEURS. However, evaluations on the FLEURS test set show that the Word Error Rate increased for the fine-tuned model on CV (23.5993% to 33.5461%), but the fine-tuned model on CV and FLEURS showed better performance than the fine-tuned model only on CV with a WER of 28.7083%.

Similar trends are observed for CER. Evaluations on the CV test set show that CER increases as we fine-tune it on more data values from 10.4123% for the pre-trained Whisper model to 10.595% for the model fine-tuned on CV and 10.8559% for the model fine-tuned on CV and FLEURS. We also observe that evaluations on the FLEURS test set show that CER increased for the fine-tuned models on CV (9.0974% to 13.51%), but the fine-tuned model on CV and FLEURS showed better performance than the fine-tuned model only on CV with a CER of 10.2842%.

When evaluated on the CV test set, spacing accuracy decreases from the pre-trained model (97.286%) to the model fine-tuned only on CV (97.0251%). The model fine-tuned on CV and FLEURS slightly lower but comparable performance on spacing with a result of 97.2599%. When evaluated on the FLEURS test set, spacing accuracy improves from 96.9985% to 96.0733%. The model fine-tuned on both CV, and FLEURS outperforms both the pre-trained and CV-fine-tuned models with a spacing accuracy of 97.211%.

4.3 Industry Model Results

We first note that IBM Watson does not support Vietnamese transcription, and, as such, only Korean results are included in the subsequent table.

	Google (vi-VN)	Google (ko-kR)	IBM Watson (ko-KR)
WER	CV: 36.3747	CV: 42.3968	CV: 67.4723
CER	CV: 16.8274	CV: 19.2328	CV: 29.5668
Spacing	CV: 96.5998	CV: 95.0678	CV: 92.5104

Table 4: Results for Industry Models

While our results in the previous section showed that our models fine-tuned on CV or CV and FLEURS did not always outperform OpenAI's pre-trained Whisper model, we find through the experimentation in this section that our fine-tuned models are competent in comparison to other industry standard models. In particular, we find that Whisper fine-tuned on CV performs better than

Google's Speech-to-Text model for both Vietnamese and Korean in terms of the WER and CER, and that margins for WER are nearly at ten percent. Performance of the IBM Watson model mostly does not seem comparable with the pre-trained Whisper or our fine-tuned Whisper models, as it showed a WER of 67.4723%, a CER of 29.5668%, which is more than double the WER and CER observed in our fine-tuned models with CV and CV and FLEURS.

5 Discussion

While fine-tuning on both the FLEURS and the CV led to an overall increased ability in transcription of Vietnamese language from text to speech, the performance increased was very minimal. This can raise important considerations about the dialectical diversity in speech recognition models. Vietnamese, being tonally and regionally diverse, presents unique challenges, as the Northern and Southern Vietnamese dialects can lead to very different pronunciations of certain common words. For example, in the Northern dialect, typically spoken around Hanoi, "vui", which means "happy," is pronounced with a rising tone, with the "v" being more pronounced. However, in the Southern dialect, commonly spoken in Ho Chi Minh City (formerly Saigon), the pronunciation of "vui" has a flatter tone, with the starting "v" sounding more like the English "j".

These tonal differences can be an issue for low-resourced languages like Vietnamese if the datasets available are not representative of both Northern and Southern Vietnamese dialects. As Hanoi is the capital of Vietnam, the northern dialect is sometimes considered "standard", leading to an under-representation of Southern Vietnamese dialect in these datasets. As a result, when datasets do contain Southern Vietnamese accents, the model will underperform. Our findings, although, show that fine-tuning increases the model's ability to transcribe Vietnamese, further improvements can be a result of ensuring linguistic equity in which there is representative inclusion of all dialects in speech recognition. Further, future works can include fine-tuning models on dialect-specific datasets and investigating their effect on performance.

Different trends between evaluation on CV and evaluation on FLEURS may result from differences in the dataset, and the model's ability to recognize speech unique to that dataset. For instance, the Common Voice dataset contains only 192 rows of Korean training data, but words included in the files are not necessarily commonly used in modern colloquial language, such as "어룽어룽하니" (which roughly translates to blurry) and "금덤판" (which roughly translates to gold mine).

We note this diversity in accents and words because characteristics specific to each dataset can be closely related to ethical considerations when building ASR systems. Speech recognition systems can exhibit biases based on training data if training data is focused exclusively on a specific subset of speakers and the words they use. The inclusion of not only multiple languages across the world but also datasets from individuals with diverse linguistic backgrounds and accents can help develop more robust, realistic, and inclusive ASR systems.

We also observe that Common Voice has 4 recorded hours of Korean and 19 recorded hours of Vietnamese. In contrast, there are 3,209 recorded hours of English and 1,067 recorded hours of French. Because we experimented with low-resource languages, future research on what evaluation metrics are for higher resource languages such as French may yield very different results for fine-tuning. From our findings, we conclude that fine-tuning is not enough for low resource languages with our setup of datasets and that developing cross-lingual models may help overcome current deficiency in resource.

6 Contribution Statement

Sophia Kang: Code for Korean segmentation rate, final report (Abstract, Related Work, Proposed Approach, Results for Korean & Industry Models, Discussion, References)

Phuc Duong: Training the models on HPC cluster, evaluation code setup, transcribing for Google and IBM Watson, final report (Results for Vietnamese, Discussion), README for GitHub repository

References

[1] Ardila, Rosana, et al., "Common Voice: A Massively-Multilingual Speech Corpus," in *Proceedings of the Twelfth Language Resources and Evaluation Conference, Association for Computational Linguistics (ACL)*, 2020. Available: https://aclanthology.org/2020.lrec-1.520/.

[2] Conneau, Alexis, et al., "FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech," 2022. Available: https://arxiv.org/pdf/2205.12446.pdf

[3] Gandhi, Sanchit, "Fine-Tune Whisper For Multilingual ASR with Transformers," 2022. Available: https://huggingface.co/blog/fine-tune-whisper.

[4] Liu, Zoey, et al., "Studying the impact of language model size for low-resource ASR," in *Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages, Association for Computational Linguistics (ACL)*, 2023. Available: https://aclanthology.org/2023.computel-1.11.pdf.

[5] OpenAI, "Introducing Whisper," 2022. Available: https://openai.com/research/whisper.

[6] Radford, Alec, et al., "Robust Speech Recognition via Large-Scale Weak Supervision," in *Proceedings of the 40th International Conference on Machine Learning (PMLR)*, 2023. Available: https://proceedings.mlr.press/v202/radford23a/radford23a.pdf.

[7] Ranathunga, Surangika, et al.,‘"Neural Machine Translation for Low-resource Languages: A Survey," in *ACM Computing Surveys*, 2023. Available: https://dl.acm.org/doi/pdf/10.1145/3567592.

[8] Singh, Satwinder, et al., "A Novel Self-training Approach for Low-resource Speech Recognition," in *Interspeech*, 2023. Available: https://www.isca-speech.org/archive/pdfs/interspeech_2023/singh23b_interspeech.pdf.

Reproducibility checklist:

Mathematical Setting, Algorithm, and Model Description:

- ☑ ~~Include a thorough explanation of the mathematical framework, algorithmic approach, and the model's architecture.~~
- ☑ ~~Ensure clarity in the methodology and theoretical underpinnings, as needed.~~

Source Code Accessibility:

- ☑ ~~Provide a link to the source code on github.~~
- ☑ ~~Ensure the code is well-documented~~
- ☑ ~~Ensure that the github repo has instructions for setting up the experimental environment.~~
- ☑ ~~Clearly list all dependencies and external libraries used, along with their versions.~~

Computing Infrastructure:

- ☑ ~~Detail the computing environment, including hardware (GPUs, CPUs) and software (operating system, machine learning frameworks) specifications used for your results~~
- ☑ ~~Mention any specific configurations or optimizations used.~~

Dataset Description:

- ☑ ~~Clearly describe the datasets used, including sources, preprocessing steps, and any modifications.~~
- ☑ ~~If possible, provide links to the datasets or instructions on how to obtain them.~~

Hyperparameters and Tuning Process:

- ☑ ~~Detail the hyperparameters used and the process for selecting them, including any search strategies like grid or random search.~~
- ☑ ~~Provide rationale for hyperparameter choices, if applicable.~~

Evaluation Metrics and Statistical Methods:

- ☑ ~~Clearly define the evaluation metrics and statistical methods used in assessing the model.~~
- ☑ ~~Include details on how these metrics are calculated.~~

Experimental Results:

- ☑ ~~Present a comprehensive set of results, including performance on test sets and/or any relevant validation sets.~~
- ☑ ~~Include comparisons with baseline models and state-of-the-art, where applicable.~~

Random Seed Reporting:

- [x] ~~If applicable, state the random seeds used in experiments to ensure reproducibility of results.~~

Ethical Considerations and Limitations:

- [x] ~~Discuss any ethical considerations related to the dataset or model use.~~
- [x] ~~Clearly state the limitations of your approach and potential areas for future work.~~