ANALYSIS OF WHISPER AUTOMATIC SPEECH RECOGNITION PERFORMANCE ON LOW RESOURCE LANGUAGE

Riefkyanov Surya Adia Pratama[1]; Agit Amrullah[2*]

Informatika[1,2]
Universitas AMIKOM Yogyakarta, Sleman, Indonesia[1,2]
www.amikom.ac.id[1,2]
rsurya@students.amikom.ac.id[1], agit@amikom.ac.id[2*]

(*) Corresponding Author

Abstract*—Implementing Automatic Speech Recognition Technology in daily life could give convenience to its users. However, speeches that can be recognized accurately by the ASR model right now are in languages considered high resources, like English. In previous research, a few regional languages like Javanese, Sundanese, Balinese and Btaknese are used in automatic speech recognition. This research aim is to improve speech recognition using the ASR model on low-resource language. The dataset used in this research is the Javanese dataset specifically because there is a high-quality Javanese speech dataset provided by previous research. The method used is fine-tuning the Whisper model which has been trained on 680,000 hours of multilingual voice data using a Javanese speech dataset. To reduce computation requirements, parameter efficient fine-tuning (PEFT) implemented in the fine-tuning process. The trainable parameter is reduced to <1% because the implementation of PEFT reduces the computation required by the model for fine-tuning. The best WER evaluation result is 13.77%, achieved by the fine-tuned Whisper large-v2 model compared to the base model of Whisper large-v2, which achieves 89.40% in WER evaluation. Performance improvement in WER evaluation showed that fine-tuning effectively improves the performance of the Whisper automatic speech recognition model on recognizing speeches in low-resource languages like the Javanese language compared to the Original Whisper model performance with minimal computational cost needed for fine-tuning large model.*

Keywords: *automatic speech recognition, low-resources language, whisper fine-tuning.*

Abstrak—*Implementasi dari teknologi Automatic Speech Recognition (ASR) pada kehidupan sehari-hari dapat memberikan kemudahan bagi para penggunanya. Namun, suara ucapan yang dapat dikenali dengan akurat oleh model ASR saat ini adalah suara ucapan dengan bahasa-bahasa sumber daya besar seperti bahasa inggris. Pada penelitian sebelumnya pengenalan suara telah dipergunakan pada beberapa bahasa daerah baik Jawa, Sunda, Bali dan Batak. Penelitian ini bertujuan untuk melakukan peningkatan penengenalan suara ucapan pada model ASR pada bahasa bersumber daya rendah. Dataset yang digunakan pada penelian ini secara spesifik adalah dataset Bahasa jawa karena terdapat dataset ucapan berbahasa jawa yang berkualitas tinggi yang disediakan oleh sebuah penelitian sebelumnya. Metode yang digunakan adalah fine-tuning pada model Whisper yang telah dilatih pada 680,000 jam data suara multilingual dengan menggunakan dataset ucapan berbahasa Jawa. Untuk mengurangi kebutuhan sumber daya komputasi, diimplementasikan parameter efficient fine-tuning (PEFT) pada proses fine-tuning. Trainable parameter dari model berkurang menjadi <1% sebagai hasil dari implementasi PEFT yang mana mengurangi kebutuhan sumber daya komputasi untuk fine-tuning model. Hasil evaluasi Word Error Rate (WER) terbaik adalah 13.77% pada model Whisper large-v2 yang telah dilakukan fine-tuning dibandingkan pada model Whisper large-v2 tanpa fine-tuning yang mana WER adalah 89.40%. Peningkatkan performa pada evaluasi WER menunjukan bahwa fine-tuning efektif untuk meningkatkan pengenalan suara ucapan otomatis model Whisper pada bahasa besumber daya rendah seperti bahasa Jawa dibandingkan dengan performa* *model Whisper aslinya dengan kebutuhan komputasi seminimal mungkin untuk model yang besar.*

***Kata Kunci**: pengenalan ucapan otomatis, bahasa bersumber daya rendah, whisper fine-tuning.*

INTRODUCTION

In recent years, Artificial intelligence technology has been developing and helping in many life activities since it was first known in 1956 (Zhang & Lu, 2021), and based on a report by (Zhang et al., 2022), publication related to AI research topic growing from 200000 in 2010 to 496010 total research publications in 2021. The Automatic Speech Recognition is one of many fields that have become the focus of various research (Alharbi et al., 2021). Automatic speech recognition (ASR) is converting speech directly from speech to word sequence using a specific algorithm using a computer (Kumar & Mittal, 2019). Several implementations of ASR technology in everyday life are internet surfing or browsing using speech voice, speech recognition, operating IoT devices using speech voice, and general human-machine interaction based on user speech voice, biometric media, *et cetera* (Zhang et al., 2019).

Two major ASR models were used and researched: the hybrid and end-to-end (E2E) models (Li, 2022). There is a transition from a widely used hybrid model to E2E because E2E models have proved more efficient than the others. The E2E model works by directly mapping input sequences to word sequences. There is three technique that is considered successful in the implementation of the E2E model: Connectionist Temporal Classification (CTC), Attention-based Encoder-Decoder (AED), and Recurrent Neural Network Transducer (RNN-T). One State-Of-Art model that implements those E2E techniques is Whisper by OpenAI.

Web-scale supervised Pre-Training For Speech Recognition (Whisper) is a model that builds based on transformer encoder-decoder architecture (Vaswani et al., 2023) and is trained in weakly supervised on 680,000 hours of multilingual and multitasking (speech recognition, translation, and language identification) speech data (Radford et al., 2022). However, as mentioned in (Radford et al., 2022), Whisper's performance could be better in recognizing language categorized as low-resource. Low-resource language is a language that has little data in the form of digital, or that can be processed by a computer directly. One example of this language category is the Javanese language (Butryna et al., 2020). Whisper performance evaluated using Character Error Rate (CER)/Word Error Rate (WER) on this type of language is remarkably low.

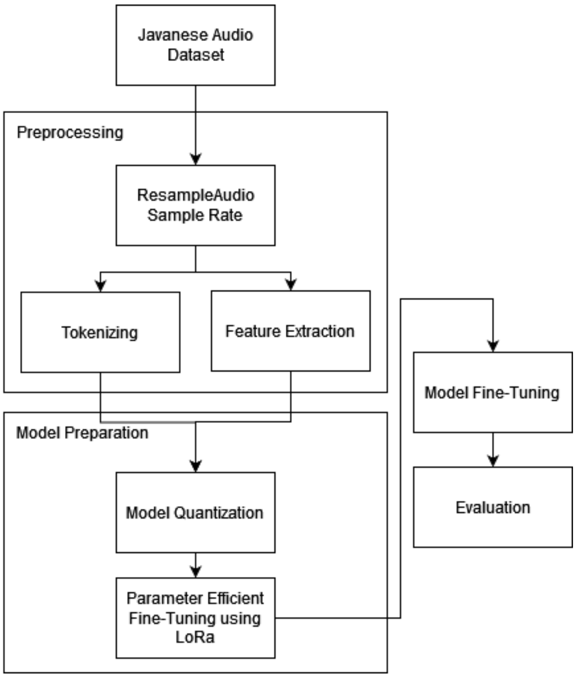
Research by (Rouditchenko et al., 2023) stated a comparison of performance between the XSL-R and Whisper model in zero-shot conditions (without fine-tuning) where the evaluation of model performance is lower in less seen or unseen language, which can categorized as low-resource language. There is research (Novitasari et al., 2020) to build an ASR model for ethnic languages in Indonesia. One of the best results is evaluating ASR model performance in recognizing speech in the Javanese language, which is 20.20% in CER evaluation.

The solution proposed to improve Whisper performance in low-resource language, mainly Javanese, is fine-tuning the Whisper model. Fine-tuning improves the AI model in specific downstream tasks (Liu et al., 2022; Min et al., 2022; Wei et al., 2022; Yang et al., 2023). Fine-tune is done by training the model using a specific dataset, which, in this case, is a dataset of speech audio that is spoken using the Javanese language, so the model is more familiar with the pattern of data. There are some problems in fine-tuning large language models like Whisper, one of which is that fine-tuning requires immense computation power. To reduce computation costs in the fine-tuning process, the Parameter Efficient Fine-Tuning (PEFT) (Fu et al., 2022) method will reduce trainable parameters trained during the fine-tuning process.

This research scope focuses on fine-tuning the Whisper ASR model in one language categorized as a low-resource language, i.e., the Javanese language. Therefore, the research aims to improve Whisper ASR's performance that WER evaluated in the Javanese language, categorized as a low-resource language. Hopefully, this research could be used as a reference by other researchers to improve ASR performance in low-resource language.

MATERIALS AND METHODS

This research aims to improve the Whisper ASR model for the Javanese language through fine-tuning using low computation. The first step is to convert the audio sample rate in the dataset from 48kHz to 16kHz to achieve maximal performance in the fine-tuning process. Then, preprocessing data is split into two stages: tokenization and feature extraction. After preprocessing, the model was prepared to compute minimal computation cost using PEFT Low-Rank Adaptation (LoRA) to reduce its trainable parameter. After all preparation is set, the Fine-tuning process could be commenced. After fine-tuning is complete, Model performance will evaluated using WER evaluation and compared with the zero shot (without fine-tuning) model. The details of the research flow can be seen in Figure 1.



Source: (Research Result, 2023)

Figure 1. Research Flow

Data Collection

The dataset will be used from (Butryna et al., 2020), which contains compilations of speech audio with .wav extension, file ID, and transcription of each audio file. There is a total of 5822 audio files with their ID and transcription, with an average duration of each audio being 3 seconds, and the total duration of all audio files is plus minus 4.85 hours. The dataset will be split into two parts, one for fine-tuning or training models and the other for evaluating model performance. Based on (Joseph, 2022), the most optimal ratio for dataset splitting is 8:2, with 8 being the training dataset and 2 being the evaluation dataset.

Audio Resampling

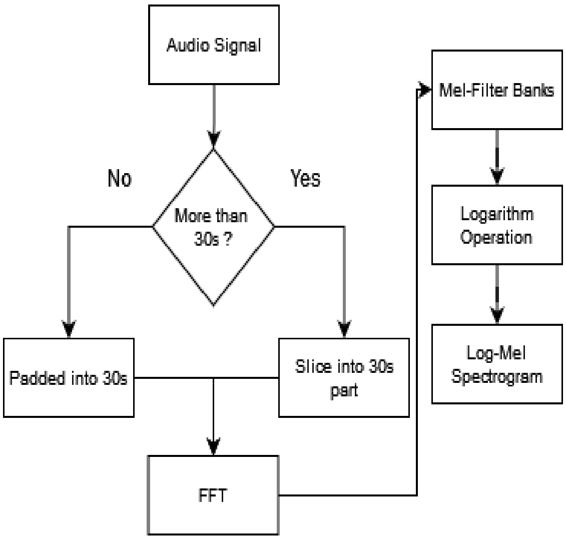
The sample rate audio in the dataset that will be used is 48kHz. Based on (Radford et al., 2022), Whisper, especially its feature extractor, works best at a 16kHz sample rate. So, to achieve maximal performance in the fine-tuning process, Audio needs to be resampled from 48kHz to 16kHz. Audio resample done using bandlimited sinc interpolation with the help of torch audio library.

Tokenization

Tokenization significantly affects language model performance even more in low-resource languages (Toraman et al., 2023). Whisper's built-in Byte-level BPE tokenizer will be used, separating sentences into words and words into tokens (Radford et al., 2019; Wang et al., 2019). There is 96 language that are recognized by Whisper's built-in tokenizer, one of which is Javanese.

Feature Extraction

Feature extraction will be done using Whisper's built-in feature extractor. The whisper feature extractor works in two steps. First, the whisper feature extractor pads the audio signal to match the 30-second duration. If audio exceeds the 30-second limit, then audio will be sliced up into two parts. If the audio signal is less than the 30-second limit, then the audio signal will padded with zero or silence into 30 seconds duration. The next step is converting audio into an 80-channel log-mel spectrogram visual. The audio signal will first be converted into a Fast Fourier Transform (FFT) measurement. Then, this measurement will be inserted into the mel-filter bank and apply the logarithm operation. The result is a visualization of the log-mel spectrogram. The workflow of Whisper Feature Extraction can be seen in Figure 2.

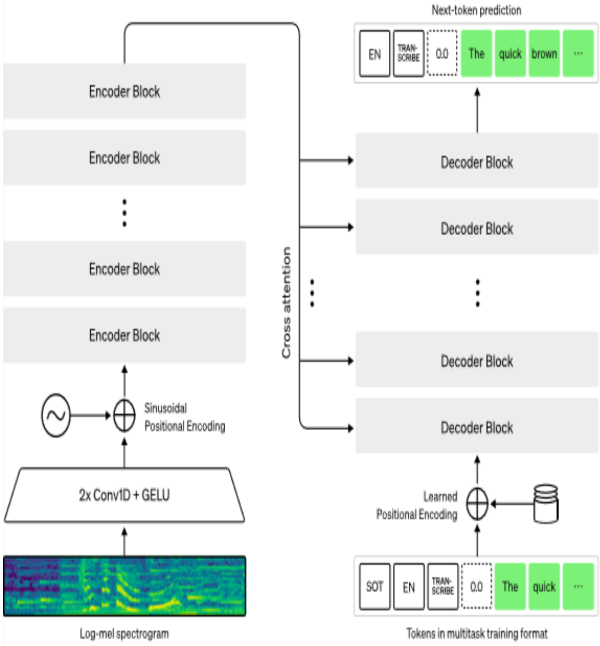


Source: (Research Result, 2023)

Figure 2. Whisper Feature Extractor Process

Model Quantization

Whisper is a model built based on an encoder-decoder transformer (Vaswani et al., 2023) or could be called a seq-2-seq model because it works by mapping input sequences into word sequences. Weak supervision training on 680,000 hours of multilingual audio has been done prior. As can be seen in Figure 3, the Whisper model will first encode the log-mel spectrogram from the Whisper feature extractor to form a sequence from the encoder's hidden state and then insert it into the decoder to predict the text token autoregressively based on the previous token condition and then encoder hidden state.



Source: (Radford et al., 2022)

Figure 3. Whisper Architecture

Whisper variant model can be seen in Table 1. For this research, models that will be used are Whisper base, Small, Medium, and Large V2.

Table 1. Whisper Model Comparison

Model	Layer	Width	Heads	Size
Tiny	4	384	6	39 M
Base	6	512	8	74 M
Small	12	768	12	244 M
Medium	24	1024	16	769 M
Large-V2	32	1280	20	1550 M

Source: (Radford et al., 2022)

Fine-tuning models with parameters that exceed 200 M requires immense computational cost. The Free computational power Google Colaboratory provides is 14.7 GB T4 GPU and 12.7 GB RAM. To address this problem, model quantization is proposed. Model quantization first proposed by (Dettmers et al., 2022) is 8-bit quantization. The full model works in full precision using the fp32 data type. The model will be loaded in quarter precision or 8-bit data type to reduce memory and computational requirements and proved that only ~5% loss in model accuracy compared to the full precision model.

Parameter-Efficient Fine-Tuning

To reduce even more computational and memory requirements, Parameter-Efficient Fine-Tuning (PEFT) is proposed. PEFT works by freezing model parameters, so only required parameters related to specific tasks will be trained (Liu et al., 2022). This reduction is possible because of the LoRA method. Low-rank Adaptation (LoRA) is a method proposed by (Hu et al., 2021) that works by adding a decomposition matrix (Update matrix) to model weights and training only newly added weights, resulting in the reduction of trainable parameters and consequently reducing memory and computational cost.

Model Fine-Tuning

Fine-tune done in Google colaboratory environment with system specification 14.7GB T4 GPU and 12.7GB RAM. The hyperparameter that used for fine-tuning can be seen in Table 2.

Table 2. Hyperparameter

Hyperparameter	Value
Learning rate	1e-3
Epoch	10
Batch_size	8
Optimizer	AdamW
Evaluation Strategy	Epoch
Gradient_Checkpointing	True

Source: (Research Result, 2023)

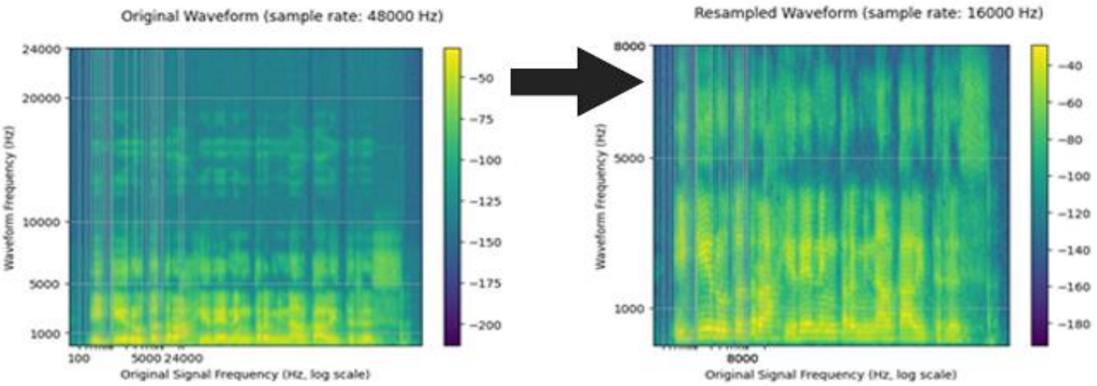
Evaluation

Word Error Rate (WER) is the evaluation matrix used to evaluate the model. WER calculation is the same as the Character Error Rate (CER). Nevertheless, WER represented ASR performance more accurately because the error rate was evaluated based on words instead of characters. Therefore, it can be said that when ASR is evaluated from WER, its error rate is higher than one thst evaluated with CER. In previous research (Novitasari et al., 2020), the evaluation used to evaluate the ASR model is CER. This method is used because the language contains some characters outside the standard alphabet. But in Javanese, it can be written in the standard alphabet. So, the WER evaluation will be used to provide a more accurate evaluation of ASR performance in recognizing Javanese speech. Mathematically, WER can be calculated with the equation (1) below where S is substitution, D is deletion, I is insertion, and C is correct.

$$WER = \frac{(S + D + I)}{(S + D + C)} \quad \text{...} (1)$$

RESULTS AND DISCUSSION

Whisper, especially its feature extractor, works best at a 16 kHz sample rate after the original sample rate of 48 kHz resampling to 16 kHz; the result of audio resampling can be seen in Figure 4.



Source: (Research Result, 2023)

Figure 4. Sample Rate Waveform Comparison

As shown in Figure 4, the original waveform visualization with a 48kHz sample rate is represented in 24kHz waveform frequency on the Y-axis and 24kHz original signal frequency on the X-axis. Compared to the original audio sample rate, the 16kHz resampled waveform in Figure 4 is represented with 8kHz waveform frequency and 8kHz original Signal Frequency, a zoomed version of Figure 4. The difference between the two sample rate visualizations is an artifact seen in the upper 8kHz of the original sample rate in Figure 4. This artifact removal does not affect speech in audio because the main speech audio is represented in the 8kHz Y-axis and X-axis range.

At the data splitting, the ratio of splitting used is 8:2, resulting in 4657 training data and 1165 evaluation data in a random state, which means that every epoch where 4657 data have been trained will be evaluated with 1165 data.

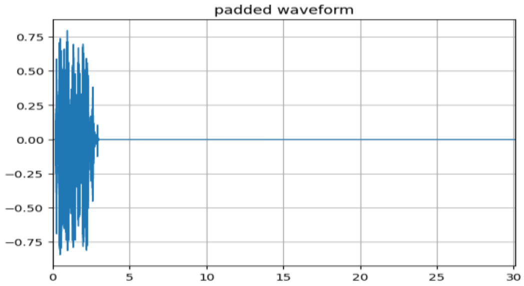
Example of the result from tokenizing with Whisper tokenizer can be seen in Table 3.

Table 3. Result of Tokenizing

Transcription	Tokenized
bar ngepeki sayuran banjur ditawake neng bandungan	{bar, ngepeki, sayuran, banjur, ditawakake, ning, bandungan}
nyebrang menyang ketapang adoh	{nyebrang, menyang, ketapang, adoh}
Puding ingkang didamel purimas radi mambet	{pudding, ingkang, didamel, purimas, radi, mambet}

Source: (Research Result, 2023)

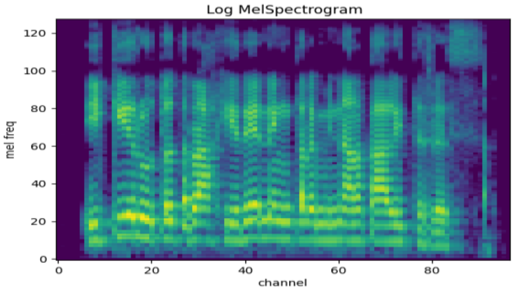
In Table 3, the Whisper tokenizer successfully tokenizes transcription text in the Javanese language into tokens per word. Because all audio data duration is less than 30 seconds, the Whisper feature extractor will pad audio waveform with 27 seconds of 0 signal or silence. The audio waveform after padding can be seen in Figure 5. In Figure 5, it can be seen that 27 seconds of silence was successfully added after the speech ended in 3 seconds.



Source: (Research Result, 2023)

Figure 5. Padded Audio Waveform

The next step is to convert this waveform into an 80-channel log-mel spectrogram visualization. The result of visualization can be seen in Figure 6.



Source: (Research Result, 2023)

Figure 6. Log -mel Spectrogram

The log-mel spectrogram was successfully visualized, and as can be seen, the log-mel spectrogram was divided into 80 channels on the X-axis. The silence is automatically omitted, and detected audio is converted into a log-mel spectrogram; as can be seen, visualization is almost identical to the waveform spectrogram in Figure 5.

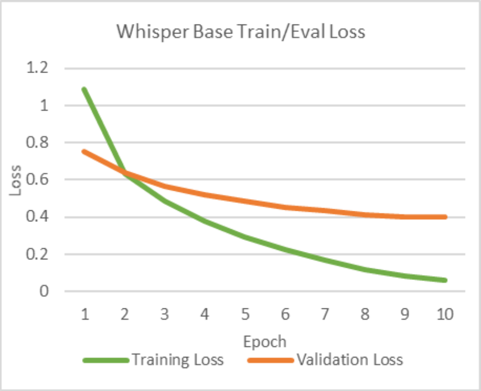
After all preprocessing steps are complete, the research will proceed into model preparation. First, we load the model in 8-bit quantization, then apply PEFT using LoRa to reduce trainable parameters. The result of the reduction can be seen in Table 4.

Table 4. Quantization and PEFT result

Model	Base Model Param.	LoRa Param.	Percentage %
Base	73,183,744	589,824	0.80
Small	243,504,384	1,769,472	0.72
Medium	768,576,512	4,718,592	0.62
Large-V2	1,551,169,280	7,864,320	0.51

Source: (Research Result, 2023)

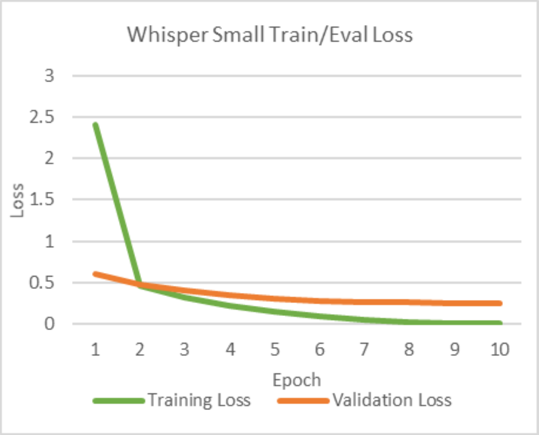
After the model is prepared, the fine-tuning process is ready to start. Figures 7, Figure 8, Figure 9, and Figure 10 presented a graph of training and evaluation loss of each model during the fine-tuning process.



Source: (Research Result, 2023)

Figure 7. Whisper Base Training/Eval Loss

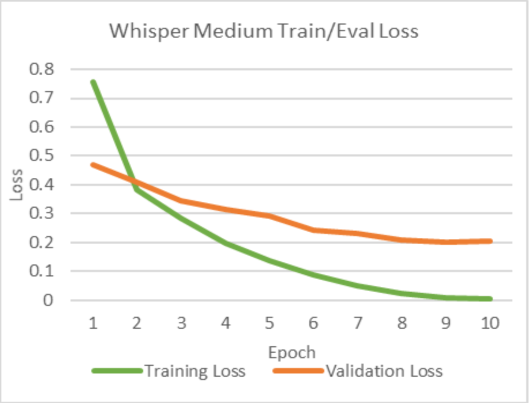
As shown in Figure 7, training and evaluation loss is reduced as the epoch continues for the Whisper-Base model. Significant improvement happened in epoch two as the model became more familiar with the pattern of Javanese speech data. There is no indication of overfitting or underfitting based on the gap between training and evaluation loss value.



Source: (Research Result, 2023)

Figure 8. Whisper Small Training/Eval Loss

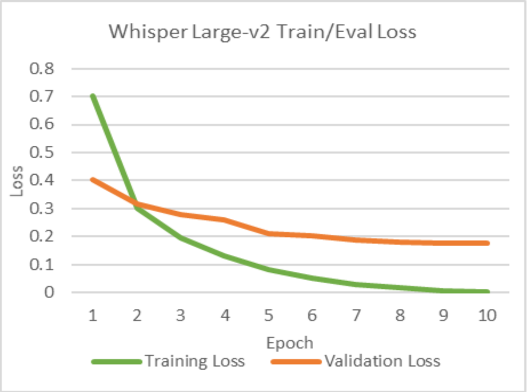
Figure 8 shows a drastic reduction in training loss in epoch two and measures up to evaluation loss. There is no significant reduction from epoch three to ten. Otherwise, there is no indication of overfitting or underfitting in the Whisper-Small model fine-tuning process.



Source: (Research Result, 2023)

Figure 9. Whisper Medium Training/Eval Loss

In Figure 9, a significant reduction of training loss happened in epoch two. The evaluation loss reduction goes down smoothly, but in epoch ten, evaluation loss is increased slightly by 0.001065%. To that end, the fine-tuned model will be used on epoch nine. The training and evaluation loss graph shows no indication of overfitting or underfitting.



Source: (Research Result, 2023)

Figure 10. Whisper Large-v2 Training/Eval Loss

Like the other model, Whisper-Large-v2 training and evaluation loss during fine-tuning goes down for each epoch, as shown in Figure 10. Training loss starts to measure up with evaluation loss in epoch two. There is a slight increase of 0.001422% in evaluation loss in epoch ten. Because of that, the end model that will be used is the model on epoch nine. There is no overfitting or underfitting indication in Whisper-Large-v2 during the fine-tuning process, as shown in the graph in Figure 10.

Comparing the results from Figures 7,8,9, and 10, there is one similarity between all models: the most significant loss value reduction is at epoch 2. This happens because models start familiarizing themselves with data patterns after two epochs. All models stop improving at epoch 9. The spotted difference is on the smaller model (base and small model) loss value, as shown in Figures 7 and 8, which goes down smoothly each epoch after epoch 2. On the other hand, in Figures 9 and 10, the medium and large model shows a visible drastic loss value reduction in epochs 3, 4, 5, and 6. This proves that larger models are learning better than smaller ones.

Table 5 shows the WER evaluation result and comparison between the base whisper model (model before fine-tuned) and the fine-tuned whisper model.

Table 5. WER Comparison

Model	Word Error Rate(WER) %	
	Base Model	Fine-tuned Model
Base	116.87	**28.57**
Small	109.63	**18.84**
Medium	110.62	**15.97**
Large-V2	89.40	**13.77**

Source: (Research Result, 2023)

In Table 5, the result of the WER evaluation is consistently better for each fine-tuned model. The improvement in WER evaluation of the performance of each Whisper ASR model is up to 85% reduction in error rate compared to the model without fine-tuning. Thus, the fine-tuning process improved WER evaluation results for low-resource language, in this case, Javanese language speech recognition.

CONCLUSION

Based on the experiment conducted above, the research could be concluded that fine-tuning the whisper model on a low-resource language dataset could improve Whisper ASR model performance in recognizing speech spoken in low-resource language, which in this research case, Javanese language that measured with Word Error Rate(WER) evaluation. The improvement in the WER result is significantly better for every Whisper model tested in this research.

REFERENCE

Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., … Almojil, M. (2021). Automatic Speech Recognition: Systematic Literature Review. *IEEE Access*, *9*, 131858–131876. https://doi.org/10.1109/ACCESS.2021.3112535

Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). Llm. int8 (): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.

Fu, Z., Yang, H., So, A. M. C., Lam, W., Bing, L., & Collier, N. (2023, June). On the effectiveness of parameter-efficient fine-tuning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 11, pp. 12799-12807).

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Joseph, V. R. (2022). Optimal ratio for data splitting. Statistical Analysis and Data Mining: The ASA Data Science Journal, 15(4), 531-538.

Kumar, A., & Mittal, V. (2019). Speech recognition: A complete perspective. International Journal of Recent Technology and Engineering (IJRTE), 7(6), 78-83.

Li, J. (2022). Recent advances in end-to-end automatic speech recognition. APSIPA Transactions on Signal and Information Processing, 11(1).

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. A. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Advances in Neural Information Processing Systems, 35, 1950-1965.

Min, S., Lewis, M., Zettlemoyer, L., & Hajishirzi, H. (2021). Metaicl: Learning to learn in context. arXiv preprint arXiv:2110.15943.

Novitasari, S., Tjandra, A., Sakti, S., & Nakamura, S. (2020). Cross-lingual machine speech chain for javanese, sundanese, balinese, and bataks speech recognition and synthesis. arXiv preprint arXiv:2011.02128.

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

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

Rouditchenko, A., Khurana, S., Thomas, S., Feris, R., Karlinsky, L., Kuehne, H., ... & Glass, J. (2023). Comparison of Multilingual Self-Supervised and Weakly-Supervised Speech Pre-Training for Adaptation to Unseen Languages. arXiv preprint arXiv:2305.12606.

Wang, C., Cho, K., & Gu, J. (2020, April). Neural machine translation with byte-level subwords. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 05, pp. 9154-9160).

Butryna, A., Chu, S. H. C., Demirsahin, I., Gutkin, A., Ha, L., He, F., ... & Wibawa, J. A. E. (2020). Google crowdsourced speech corpora and related open-source resources for low-resource languages and dialects: an overview. arXiv preprint arXiv:2010.06778.

Toraman, C., Yilmaz, E. H., Şahinuç, F., & Ozcelik, O. (2023). Impact of tokenization on language models: An analysis for turkish. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(4), 1-21.

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

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., ... & Le, Q. V. (2021). Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652.

Yang, H., Zhang, M., Tao, S., Ma, M., & Qin, Y. (2023, February). Chinese ASR and NER Improvement Based on Whisper Fine-Tuning. In 2023 25th International Conference on Advanced Communication Technology (ICACT) (pp. 213-217). IEEE.

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224.

Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., ... & Perrault, R. (2021). The AI index 2021 annual report. arXiv preprint arXiv:2103.06312.

Zhang, X., Peng, Y., & Xu, X. (2019, September). An overview of speech recognition technology. In 2019 4th International Conference on Control, Robotics and Cybernetics (CRC) (pp. 81-85). IEEE.