A Comparative Study of Whisper and Wav2Vec 2.0 for End-to-End Automatic Speech Recognition in Low-Resource Settings

Jurin Vachhani, Aashi Goyani, Vinal Gadhiya

The University of Texas at Arlington

Abstract

Automatic Speech Recognition (ASR) has witnessed remarkable advancements with the emergence of transformer-based architectures such as Wav2Vec 2.0 and Whisper. This study presents a comparative analysis of these two state-of-the-art models under constrained training conditions. We fine-tuned facebook/wav2vec2-base and openai/whisper-small using preprocessed audio-text datasets and evaluated their performance based on Word Error Rate (WER) and validation loss. Wav2Vec 2.0 achieved a significantly lower WER of 10% compared to Whisper’s 34%, demonstrating stronger transcription accuracy, albeit with greater computational requirements. Whisper's lightweight training setup, involving frozen encoder layers, highlighted its potential in low-resource scenarios. Additionally, the implementation of memory optimizations, dynamic batching, and sample filtering further improved training stability and efficiency. The findings emphasize the trade-offs between computational overhead and model performance, and outline potential directions for future work, including multilingual dataset evaluation, large-scale training, and model deployment.

Introduction

The field of Automatic Speech Recognition (ASR) has rapidly evolved with the advent of transformer-based architectures, enabling significant improvements in speech-to-text systems. Among the leading approaches, Wav2Vec 2.0 and Whisper represent two state-of-the-art frameworks that leverage self-supervised pretraining and fine-tuning to build robust ASR systems. These models have been extensively studied in high-resource settings; however, their comparative performance in constrained, real-world training environments remains underexplored.

In this work, we aim to bridge this gap by empirically evaluating facebook/wav2vec2-base and openai/whisper-small under similar training conditions. Our primary focus is on scenarios with limited compute and data resources, which are often encountered in practical deployments. We investigate the impact of training strategies such as gradient accumulation, mixed precision, encoder freezing, and checkpointing, and assess each model’s performance based on standard ASR metrics including Word Error Rate (WER) and validation loss.

The study provides detailed insights into the development and training phases of both models. For Wav2Vec 2.0, we employ the Hugging Face Dataset and AutoProcessor pipeline, using Connectionist Temporal Classification (CTC) loss to train the model over 10 epochs. Whisper, on the other hand, is trained with a frozen encoder and employs WhisperFeatureExtractor and WhisperTokenizer for preprocessing. While Wav2Vec 2.0 yields stronger performance with a WER of 10%, Whisper offers faster and more memory-efficient training, albeit with a higher WER of 34%.

This paper not only contrasts the models' accuracy and efficiency but also discusses practical enhancements such as memory optimization and sample filtering. We conclude by outlining future directions, including scaling to larger and multilingual datasets and deploying the models in real-world applications as interactive demos.

Data Preparation

To ensure the quality and consistency of the training data, we applied a structured and multi-step data preparation pipeline to the English subset of the Common Voice dataset. This subset offers a large and diverse set of audio recordings, capturing various accents, speech styles, and recording conditions. However, due to its crowdsourced nature, preprocessing was essential to eliminate noise and enhance the reliability of training inputs.

As an initial filtering step, we removed any samples in which the number of downvotes exceeded the number of upvotes. This heuristic helped eliminate clips flagged by the community as inaccurate, unclear, or poorly transcribed. The remaining samples underwent a three-stage audio preprocessing sequence.

First, we performed normalization to standardize the loudness across all clips, mitigating variations introduced by differing microphones or speaking volumes. This was followed by reference sample normalization, where each audio clip's loudness was adjusted to match a predefined reference sample, further promoting consistency and reducing loudness-based bias during training. Finally, we converted all audio files from MP3 to WAV format, opting for WAV due to its lossless nature and better suitability for deep learning models that rely on high-fidelity input.

This pipeline ensured that the data used for fine-tuning was acoustically clean, uniformly processed, and aligned with the requirements of modern ASR systems. It significantly improved the models' ability to learn robust speech representations while minimizing artifacts introduced by inconsistent or low-quality audio data.

Project Description

The objective of this project is to develop an effective automatic speech recognition (ASR) system that bridges the gap between human speech and machine comprehension using state-of-the-art deep learning models. The system is designed to accurately convert spoken language into written text, thereby providing a reliable solution for speech transcription across various real-world applications.

To achieve this, we fine-tuned two prominent transformer-based architectures—Wav2Vec 2.0[2] and Whisper[1]—on a subset of the Common Voice dataset. Both models are known for their strong performance in low-resource and noisy environments, as well as their ability to handle diverse accents and speech patterns. The models were fine-tuned for 10 epochs to adapt to the specific characteristics of the dataset and improve transcription accuracy.

This project highlights the potential of modern ASR systems in achieving high-quality transcription performance by utilizing pre-trained deep learning models and large-scale, publicly available datasets.

Main References

The foundation of this project is built upon state-of-the-art pretrained models for automatic speech recognition (ASR), specifically Whisper[1] and Wav2Vec 2.0[2], both of which were accessed through the Hugging Face Transformers library[3]. Leveraging these pretrained models enabled efficient fine-tuning and rapid development of a high-performance speech-to-text system.

Whisper, developed by OpenAI, is a robust multilingual and multitask speech recognition model trained on a large-scale dataset collected from the web. It demonstrates strong performance even in noisy and low-resource conditions. In our work, we used the pretrained Whisper model from Hugging Face and fine-tuned it using cross-entropy loss. Model performance was evaluated using Word Error Rate (WER) and validation loss.

Wav2Vec 2.0, introduced by Facebook AI (now Meta AI), utilizes a self-supervised learning framework to learn contextualized speech representations from raw audio. We fine-tuned the pretrained Wav2Vec 2.0 model available on Hugging Face using Connectionist Temporal Classification (CTC) loss. Evaluation metrics included WER and validation loss.

By using Hugging Face's implementations, we were able to focus on experimentation and model performance while relying on well-maintained and widely used versions of the models. The metrics and training strategies were aligned with best practices established in the original papers, ensuring reproducibility and performance integrity.

The primary references that guided our project include the original research papers for Whisper and Wav2Vec 2.0, the Hugging Face model documentation, and community tutorials related to speech model fine-tuning and evaluation.

Difference from original paper

Whisper

This project builds upon the Whisper model developed by OpenAI, which was trained from scratch on 680,000 hours of multilingual and multitask supervised data using a highly resource-intensive setup involving 256 GPUs. The original model was designed as a general-purpose multilingual speech recognition system with robustness to noise and variability across languages and accents. Evaluation in the original work was performed on a wide range of tasks, including multilingual transcription and translation, under diverse and noisy conditions.

In contrast, our implementation focused on fine-tuning the pretrained Whisper model—made available through Hugging Face—on a domain-specific subset of the Common Voice dataset, consisting of 2,000 English audio samples. To conserve computational resources, training was conducted on a single GPU, utilizing FP16 precision and gradient checkpointing to reduce memory consumption.

While the original model was evaluated broadly, we concentrated on Word Error Rate (WER) as the primary metric, assessing performance solely on the English dataset. Moreover, to ensure a cleaner and more specialized training pipeline, we fine-tuned the model on a cleaned version of Common Voice data, which allowed us to tailor the model for improved accuracy in domain-specific transcription tasks.

This comparison highlights how modern pretrained models can be effectively adapted to resource-constrained environments and specific application domains with minimal data and compute, while still leveraging the benefits of large-scale foundational training.

Wav2Vec 2.0

Wav2Vec 2.0, developed by Facebook AI, was originally pretrained using large-scale unlabeled audio data with large batch sizes and multi-GPU setups. The original implementation relied on FP32 precision and fixed-length padding, with evaluation metrics encompassing both Word Error Rate (WER) and Phoneme Error Rate (PER), enabling comprehensive analysis across linguistic levels. The robustness of the original model was not primarily centered on handling misaligned or noisy data samples.

In our project, we fine-tuned a pretrained Wav2Vec 2.0 model, available on Hugging Face, using a curated subset of 5,000 English audio samples from the Common Voice dataset. This was achieved using a single GPU, with FP16 precision, dynamic batching, and gradient checkpointing, resulting in approximately 30% reduction in memory usage. We also incorporated a custom data collator and dynamic padding, allowing the model to efficiently handle variable-length inputs, unlike the original fixed-length batch processing.

Our evaluation was streamlined, focusing solely on WER to facilitate focused benchmarking. Furthermore, to improve robustness within our domain-specific use case, data validation mechanisms were introduced to detect and correct misaligned or problematic samples during training.

This adaptation demonstrates how large pretrained models like Wav2Vec 2.0 can be customized and optimized for constrained computing environments while maintaining performance for specific speech recognition tasks.

Model Training and Performance

Whisper

For this study, we employed the openai/whisper-small model as the base architecture for automatic speech recognition. To optimize training efficiency and reduce computational requirements, the encoder was kept frozen while only the decoder layers were fine-tuned. Feature extraction was carried out using the WhisperFeatureExtractor and WhisperTokenizer from the Hugging Face Transformers library.

The model was trained using a batch size of 4 with gradient accumulation to simulate larger batch training. Mixed precision (fp16) training was enabled to reduce memory usage and speed up computations, and training was conducted for 10 epochs, totaling approximately 2200 steps. Checkpoints were used throughout to ensure stability and allow for training resumption if necessary.

The final model achieved a Word Error Rate (WER) of 34% and a validation loss of 0.44. These results indicate that even with limited computational resources and a relatively small dataset, the Whisper-small model can be effectively fine-tuned to perform reasonably well on English speech recognition tasks.

Wav2Vec 2.0

For this study, we fine-tuned the facebook/wav2vec2-base model, a transformer-based architecture designed for self-supervised learning of speech representations. The training was conducted on preprocessed datasets formatted as .csv files and further adapted using the Hugging Face Dataset framework. Feature extraction for both audio and corresponding text labels was performed using the AutoProcessor class, enabling seamless integration with the wav2vec 2.0 architecture.

The training setup involved a batch size of 8, with gradient accumulation steps set to 4 to effectively simulate a larger batch size without exhausting GPU memory. The model was trained for 10 epochs with a learning rate of 0.003. Connectionist Temporal Classification (CTC) loss was utilized as the primary objective function, suitable for alignment-free training of speech-to-text models. To optimize performance and memory efficiency, mixed precision training was employed along with gradient checkpointing.

The final model demonstrated promising results with a Word Error Rate (WER) of 10%, indicating strong transcription accuracy. The total CTC loss after training converged to 125, suggesting stable and effective learning throughout the epochs.

Analysis

This study explored the training and performance characteristics of two state-of-the-art automatic speech recognition (ASR) models: Whisper and Wav2Vec 2.0. Both models were fine-tuned with a focus on optimizing training efficiency while maintaining acceptable transcription accuracy. The results highlight a balance between computational feasibility and model performance, particularly under limited-resource settings.

The fine-tuning of the facebook/wav2vec2-base model resulted in a notably low Word Error Rate (WER) of 10%, demonstrating high transcription accuracy. This was achieved through effective utilization of techniques such as mixed precision training, gradient accumulation, and checkpointing, which collectively allowed for efficient memory usage and stable convergence, as reflected by the final CTC loss of 125. The adoption of the Hugging Face Dataset framework and the AutoProcessor class also contributed to a seamless and robust data pipeline for training.

In the case of Whisper, the model was trained with the encoder layers frozen, which helped reduce computational load and training time. Despite this, the model achieved a moderate WER of 34% with a validation loss of 0.44. The results suggest that while Whisper is effective for lightweight applications, its performance may be constrained when trained on smaller datasets or without fine-tuning all layers.

During the fine-tuning process, we encountered several technical challenges that required custom solutions. For both Wav2Vec 2.0 and Whisper, the built-in data collator provided by Hugging Face did not adequately pad audio samples within each batch. Since ASR models expect inputs of consistent length across batches for efficient training, improper padding resulted in unstable training dynamics. To address this, we developed a custom data collator that ensured each batch consisted of properly padded audio samples, significantly improving training reliability.

This step proved critical for successful model training and prediction. Without the custom data collator, particularly for Wav2Vec 2.0, the model consistently generated only padding tokens during validation, leading to a WER of 1.0 (100%) after every epoch. As a result, the model failed to learn meaningful speech representations. After implementing the custom collator, however, we observed progressive improvement: the WER began to decrease steadily with each epoch, and the model started producing coherent and accurate transcriptions. Thus, the custom data collator was essential for enabling effective learning and achieving good ASR performance.

Furthermore, fine-tuning the Wav2Vec 2.0 model required an additional text preprocessing step that was not needed for Whisper. Wav2Vec 2.0 expects input text in uppercase, with each word separated by a special delimiter ("|"). This formatting is crucial for the model's Connectionist Temporal Classification (CTC) loss function to correctly align predictions with ground truth labels. To meet these requirements, we implemented a customized preprocessing function that automatically converted all text to uppercase and inserted the necessary delimiters before feeding the data into the training pipeline.

Comparing the current Wav2Vec 2.0 implementation to the original reference from the paper, several enhancements were introduced, including 30% memory optimization, dynamic batching support, and improved robustness through the filtering of misaligned samples. However, a limitation of this comparison lies in the scope of evaluation metrics; the original work also considered phoneme-level metrics such as PER (Phoneme Error Rate), which were not included in this study.

While the results are encouraging, there remain opportunities for further improvement. Whisper's performance could potentially benefit from training on a larger and more diverse dataset and unfreezing the encoder for deeper fine-tuning. Moreover, the evaluation of Wav2Vec 2.0 could be extended to include additional metrics for a more comprehensive analysis.

Looking forward, there are several avenues for future work. Expanding the experiments to larger datasets could significantly enhance model generalization and accuracy, allowing the models to better handle diverse speech patterns and noisy environments. Deploying the trained models as interactive web or API-based demos would increase accessibility and demonstrate their real-world applications. Additionally, incorporating multilingual datasets would assess the models' adaptability to different languages, expanding their usability across a broader range of scenarios.

Conclusion

This study presented a comparative evaluation of two leading speech-to-text models—Wav2Vec 2.0 and Whisper—fine-tuned on a subset of the Common Voice dataset under constrained resource conditions. Through systematic experimentation, we demonstrated that Wav2Vec 2.0, when paired with memory-efficient training strategies such as mixed precision and gradient checkpointing, achieved a significantly lower Word Error Rate (10%) compared to Whisper (34%). On the other hand, Whisper showed advantages in training speed and computational efficiency, making it a suitable candidate for lightweight applications, especially when training on limited data with partially frozen layers.

The implementation also included enhancements such as dynamic batching and sample filtering, which contributed to more stable and effective training. While the original implementations of these models often leveraged large-scale datasets and high compute resources, our project explored their performance in more realistic, limited settings—offering practical insights into deploying ASR systems in resource-constrained environments.

Looking forward, expanding the dataset size, exploring multilingual training, and deploying the models as interactive web demos stand as promising directions for extending this work. These steps will not only help in improving transcription accuracy and model generalizability but also open avenues for broader real-world applications across diverse linguistic and acoustic scenarios.

References:

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” OpenAI, 2022.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020.

[3] Hugging Face, “Transformers: State-of-the-art Natural Language Processing for Pytorch, TensorFlow, and JAX”

[4] Mozilla Foundation, “Common Voice: An Open Source Voice Dataset”