CS4681 - Advanced Machine Learning Short Paper

210163M

Abstract

This paper presents three experiments conducted to improve Automatic Speech Recognition (ASR) performance on the LibriSpeech dataset using the WavLM-Large model. The experiments investigate fine-tuning strategies for WavLM using a Connectionist Temporal Classification (CTC) head and explore the integration of a 4-gram KenLM language model during decoding. The first experiment establishes a baseline using 60% of the LibriSpeech train-clean-100 subset. The second experiment fine-tunes the model on the full 100-hour dataset with optimized hyperparameters, while the third experiment incorporates KenLM for context-aware decoding. The results show a consistent reduction in Word Error Rate (WER) across experiments, achieving 4.16% on the test-clean set and 3.94% on the validation set, approaching the benchmark performance of WeNet (2.7%).

1. Introduction

Automatic Speech Recognition (ASR) has evolved rapidly with the introduction of self-supervised pretraining models such as Wav2Vec 2.0, HuBERT, and WavLM. These models learn general-purpose speech representations from large amounts of unlabeled audio, enabling efficient fine-tuning on smaller labeled datasets.WavLM, developed by Microsoft Research, extends earlier models by adding denoising and speaker-aware pretraining, producing robust embeddings suitable for various downstream speech tasks.

Despite the strength of WavLM representations, downstream performance still depends heavily on fine-tuning strategy and decoding method. This study investigates how parameter tuning and decoder selection influence ASR accuracy when using WavLM-Large on LibriSpeech. The paper presents three sequential experiments that progressively refine performance and provide technical insights into each improvement stage.

2. Related Work

Self-supervised models have become the backbone of modern ASR. Wav2Vec 2.0[2] introduced masked latent prediction for audio, while HuBERT [3] extended this with clustered hidden targets. WavLM [1] further enhanced pretraining by introducing denoising tasks and

speaker-aware objectives.
The Connectionist Temporal Classification (CTC) loss [4] enables alignment-free sequence learning but lacks language-level context, often requiring a language model for decoding. KenLM [5], a fast and efficient n-gram language model, is widely used to improve decoding by incorporating word-level probabilities. WeNet [7] combined attention and CTC mechanisms with an external LM to reach a 2.7% WER on LibriSpeech, setting a strong benchmark for open-source ASR systems.

---

3. Methodology

3.1 Model Structure

The proposed system consists of three components:

1. Encoder: The pretrained WavLM-Large model, frozen during early training stages and later fine-tuned.
2. CTC Head: A linear layer mapping encoder outputs to character probabilities.
3. Decoder: Greedy decoding for baseline experiments, and KenLM-based beam search in the final stage.

3.2 Dataset

All experiments used the LibriSpeech dataset (Panayotov et al., 2015).

- Experiment 1: 60% of train-clean-100 (≈60 hours)
- Experiment 2: Full train-clean-100 (100 hours)
- Experiment 3: Full train-clean-100 with KenLM decoding
- Validation was done on dev-clean, and evaluation on test-clean.

3.3 Evaluation Metrics

Performance was measured using Word Error Rate (WER) and Character Error Rate (CER). Lower values indicate better transcription accuracy.

---

4. Experiments

4.1 Experiment 1 – Baseline on 60% of LibriSpeech-100h

Objective: Validate training pipeline and establish a baseline using a limited subset.

Training setup:

- Batch size: 4
- Learning rate: 3e-4
- Weight decay: 0.005
- Warmup steps: 100
- Scheduler: Linear
- Epochs: 3

Motivation:
The high learning rate and short warmup allow the model to adapt quickly on small data, while linear scheduling simplifies the early convergence phase.

Results:
After 3 epochs, the validation WER improved from 35% to **18.1%**, and validation loss decreased to 0.197. This confirmed effective fine-tuning but revealed that limited data restricted language coverage.

4.2 Experiment 2 – Full Dataset with Optimized Hyperparameters

Objective: Improve performance and stability through better regularization and scheduling.

Key modifications:

- Batch size increased from 4 to 8
- Learning rate reduced from 3e-4 to 2e-4
- Weight decay increased to 0.01
- Warmup steps increased to 1000
- Scheduler changed from linear to cosine
- Gradient checkpointing enabled
- Epochs increased to 4

Rationale:

- A larger batch improves gradient stability.
- A lower learning rate with a longer warmup prevents catastrophic forgetting of pretrained features.
- The cosine learning rate scheduler provides smooth convergence.
- Stronger weight decay improves generalization.
- Gradient checkpointing allows training on limited GPU memory.

Results:

Step	Train Loss	Val Loss	WER
1200	1.3196	0.5604	0.5092
2000	0.4343	0.1969	0.1734
4000	0.2041	0.1025	0.0822
6800	0.1469	0.0878	0.0682

Analysis:
The WER dropped to **6.82%**, a 62% relative improvement over Experiment 1. The longer warmup and cosine scheduler improved convergence smoothness, while weight decay reduced overfitting.

4.3 Experiment 3 – Decoder Fine-Tuning with KenLM

Objective: Integrate linguistic knowledge during decoding to correct word-level errors.

KenLM Parameters:

- Beam width: 100
- LM weight (α): 0.8
- Word insertion penalty (β): -0.2

Results:

- Validation (dev-clean): WER = **3.94%**, CER = **1.34%**
- Test (test-clean): WER = **4.16%**, CER = **1.35%**

Discussion:
KenLM decoding significantly reduced WER by capturing linguistic dependencies that CTC-based greedy decoding cannot. The model learned strong acoustic features but benefited from additional word-level context.

---

5. Discussion and Technical Validation

5.1 Why Parameter Changes Helped

- **Data scaling:** Increasing the training set improved generalization by exposing the model to diverse accents and phoneme patterns.
- **Learning rate and warmup:** A smaller learning rate and longer warmup allowed more stable fine-tuning.

- **Scheduler:** Cosine decay provided gradual learning-rate adjustments, helping the model achieve lower final losses.
- **KenLM:** The language model corrected phonetically similar but linguistically implausible outputs, such as “their” vs. “there.”

5.2 Ablation Observations

- Without gradient checkpointing, larger batch training was impossible on limited hardware.
- Removing the cosine scheduler increased validation WER by about 1.5%.
- Overweighting the LM ($\alpha > 1.5$) produced overconfident common n-grams, slightly worsening accuracy.

5.3 Comparison with WeNet

- WeNet reported 2.7% WER on LibriSpeech test-clean using a CTC-attention hybrid trained on 960 hours.
- Our system achieved **4.16% WER** using only 100 hours, demonstrating that WavLM’s pretrained representation can deliver high accuracy even with limited supervised data.

6. Limitations

- The experiments used only the LibriSpeech-100h subset, so results may not generalize to larger datasets or noisier domains.
- Beam search with KenLM introduces additional decoding latency, making it less ideal for real-time systems.
- Statistical language models can bias predictions toward frequent words, which could be mitigated by neural LM integration in future work.

7. Conclusion and Future Work

- This paper demonstrated that fine-tuning WavLM-Large with a CTC head and integrating a KenLM decoder substantially improve ASR performance on LibriSpeech.
- The progressive experiments show how dataset size, hyperparameter tuning, and decoding strategy jointly influence performance.
- The final system achieved a WER of **4.16%** on test-clean, approaching benchmark performance with less than one-eighth of the training data used in WeNet.

Future work will include scaling to LibriSpeech-460h and 960h datasets, exploring neural language model fusion, and analyzing errors across accents and speaker conditions.

8. References

1. Chen, S., Zhang, Y., Xu, K., et al. (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.

2. Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.* Advances in Neural Information Processing Systems (NeurIPS).

3. Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., et al. (2021). *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units.* IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 3451–3460.

4. Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks.* Proceedings of ICML, 369–376.

5. Heafield, K. (2011). *KenLM: Faster and Smaller Language Model Queries.* Proceedings of the Sixth Workshop on Statistical Machine Translation, 187–197.

6. Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). *LibriSpeech: An ASR Corpus Based on Public Domain Audio Books.* Proceedings of ICASSP, 5206–5210.

7. Zhang, B., Liu, R., Huang, J., et al. (2021). *WeNet: Production Oriented End-to-End Speech Recognition Toolkit.* Interspeech 2021.

8. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems (NeurIPS).

9. Loshchilov, I., & Hutter, F. (2019). *Decoupled Weight Decay Regularization.* International Conference on Learning Representations (ICLR).

10. Chen, T., Xu, B., & Zhang, C. (2016). *Training Deep Nets with Sublinear Memory Cost.* arXiv preprint arXiv:1604.06174.