A Compact Whisper+LoRA Baseline for Taiwanese Hakka ASR in FSR-2025

Hung-Ting Hsieh
Independent Researcher
hsiehk0214@gmail.com

Abstract

We present a compact baseline for the Formosa Speech Recognition (FSR-2025) Taiwanese Hakka ASR challenge. Our system fine-tunes *Whisper-large-v2* (Track 1) and *Whisper-large-v3-turbo* (Track 2) (Radford et al., 2022) with LoRA (Hu et al., 2021), using consistent text normalization and balanced dev splits, without external data or language models. On the official warm-up set, we obtain 10.94% CER for Track 1 (Hanzi) and 28.48% SER for Track 2 (Pinyin). We provide simple yet reproducible pipelines covering data preparation, training, inference, and evaluation. Code is available at github.com/Kevindic0214/FSR-Challenge-2025.

Keywords: Automatic speech recognition; Hakka; Whisper; LoRA; low-resource; FSR-2025; CER; SER

1 Introduction

Taiwanese Hakka is a low-resource language variant of significant cultural value. FSR-2025 defines two tracks: Track 1 evaluates character error rate (CER) on Hanzi, and Track 2 evaluates syllable error rate (SER) on Pinyin. We aim to provide a strong, minimal-requirement baseline using *Whisper-large-v2* (Track 1) and *Whisper-large-v3-turbo* (Track 2) fine-tuned with low-rank adaptation (LoRA), emphasizing practical engineering choices and reproducibility over model complexity.
In this work, we follow the official specification of the FSR-2025 challenge (FSR2025).

Contributions. (i) A reproducible baseline for both tracks with unified normalization that matches evaluation; (ii) a simple LoRA recipe runnable on a single 24 GB GPU with balanced speaker-based dev split; (iii) competitive warm-up results with lightweight error and length-bucket analyses.

Table 1: Dataset overview and evaluation split.

Split	Size	Notes
HAT-Vol2 (train)	∼60 h	Dapu/Zhao'an; 16 kHz mono
Warm-up (eval)	∼10 h / 4,299 utt	Official FSR-2025 set
Dev speakers	12 (balanced)	DF/DM/ZF/ZM allocation

2 Task and Data

We train on the HAT-Vol2 corpus (∼60 hours; Dapu and Zhao'an dialects; 16 kHz mono) and evaluate on the FSR-2025 warm-up set (∼10 hours; 4,299 utterances total). We build manifests via dedicated scripts for each track, apply Unicode NFKC normalization, remove zero-width characters, and adopt track-specific text processing: Hanzi cleaning for Track 1 and Pinyin digit-tone policy for Track 2. Dev speakers are selected in a balanced way across DF/DM/ZF/ZM groups for stable validation. We rely on the HAT-Vol2 dataset (HAT-Vol2) and the official warm-up set (FSR2025) for training and evaluation.

Normalization policy. Track 1 (Hanzi): apply NFKC, remove zero-width characters, map mixed punctuation to Chinese forms, and strip spaces to align with evaluation. Track 2 (Pinyin): apply NFKC, remove zero-width characters, map ü/ǘ/… and "u:/U:" to "v", keep only [a–z0–9] and single spaces, and by default drop starred syllables (e.g., "*ki53" or "ki53*"); an optional fix merges split-tone forms (e.g., "ki 53" → "ki53").

Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025), pages 467–470
November 20-22, 2025

3 Related Work

Low-resource ASR has been explored in multilingual programs such as Babel (Harper, 2014). Whisper (Radford et al., 2022) is a strong multilingual recognizer; we adapt it to Hakka via parameter-efficient fine-tuning. LoRA (Hu et al., 2021) reduces trainable parameters for seq2seq models while retaining quality, enabling practical fine-tuning on 24 GB GPUs.

4 Approach

We fine-tune *Whisper-large-v2* (Radford et al., 2022) with LoRA (Hu et al., 2021) (rank 16, α=32, dropout 0.05). Training uses gradient checkpointing, bf16 when available, and label smoothing. For Track 1 decoding, we force Chinese transcription via the decoder prompt; Track 2 uses language-appropriate decoding without language forcing. Beam search with 5 beams and temperature 0.0 is used unless specified.

Implementation details: we apply LoRA adapters to attention and MLP modules (`q_proj`, `k_proj`, `v_proj`, `out_proj`, `fc1`, `fc2`); enable TF32 for faster, stable training on recent GPUs; and use label smoothing of 0.1.

We keep Whisper's default suppression behavior (do not forcibly clear `suppress_tokens`), disable the generation cache during training, and enable early stopping on the dev metric (patience 2). bf16 is automatically used when supported; otherwise fp16 on GPU.

Implementation. We implement training and inference with HuggingFace Transformers and PEFT on PyTorch. Audio I/O uses `torchaudio` for Track 1 and `soundfile`+`librosa` for Track 2. Manifests are JSONL with fields {`utt_id`, `audio`, `text/hanzi/pinyin`, `group`}; relative audio paths are resolved via a root flag. During training we enable gradient checkpointing (non-reentrant when available), set `use_cache`=False, and turn on TF32. Decoding uses `num_beams`=5, `temperature`=0.0, `no_repeat_ngram_size`=3, `length_penalty`=1.0, and `max_new_tokens`=256; we force a Chinese decoder prompt only for Track 1. We log CER/SER, exact match, group/length-bucket scores, 3-gram repetition rate, throughput, and peak memory; training uses AdamW with a linear schedule and 500 warmup steps, and we save only the LoRA adapter and processor for lightweight deployment.

Track	Metric	Score
Track 1 (Hanzi)	CER / EM	10.94% / 58.06%
Track 2 (Pinyin)	SER / EM	28.48% / 12.17%

Table 2: Warm-up evaluation results. EM: exact match.

5 Experiments

We train for 3 epochs with per-device batch size 2 and gradient accumulation 16 on an RTX 4090 (24 GB). Evaluation metrics are CER (Track 1) and SER (Track 2) with sentence-level exact match for reference. We use seed 1337, learning rate 1×10^{-4} with 500 warmup steps, label smoothing 0.1, gradient checkpointing, TF32, and early stopping (patience 2). The HuggingFace Trainer default optimizer (AdamW) is used.

6 Results

On the warm-up set: Track 1 reaches 10.94% CER with 58.06% exact match; Track 2 reaches 28.48% SER with 12.17% exact match. These numbers are obtained with the shared pipelines and no external data beyond the provided corpora. We observe stable validation under balanced speaker splits and consistent normalization. Longer utterances show mildly higher error rates (notably in the 12.4–20 s bucket), and we observe small variations across DF/DM/ZF/ZM groups under the balanced-split protocol.

Final-test results. On the official final-test, our system achieves 18.78% CER for Track 1 (ranked 2/3 in our social group) and 33.38% SER for Track 2. For analysis, we also report a tone-removed Pinyin WER of 21.30% (ranked 2/2 among teams with available data). Figure 2 shows the official charts.

7 Reproducibility

Code and scripts are available at github.com/Kevindic0214/FSR-Challenge-

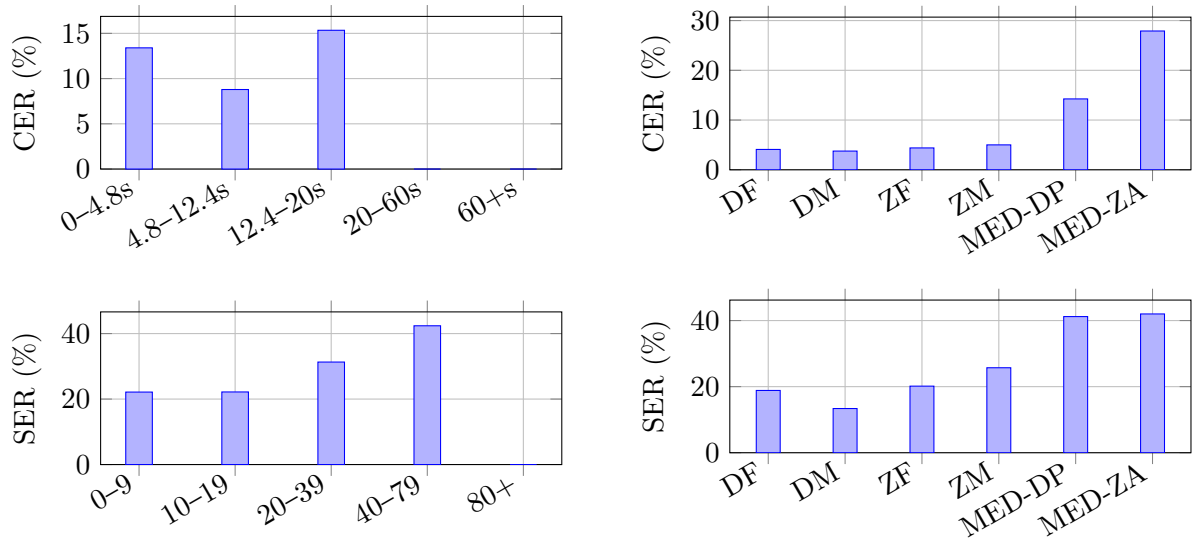

Figure 1: Warm-up analysis: (a) Track 1 CER by utterance duration; (b) Track 1 CER by group; (c) Track 2 SER by syllable length; (d) Track 2 SER by group.

Set	Track	Metric	Score (rank)
Final-test	1 (Hanzi)	CER	18.78% (2/3)
Final-test	2 (Pinyin)	SER	33.38% (2/2*)
Final-test	2 (Pinyin)	WER (tone-removed)	21.30% (2/2*)

Table 3: Official final-test results. *Among teams with available data in our social group.

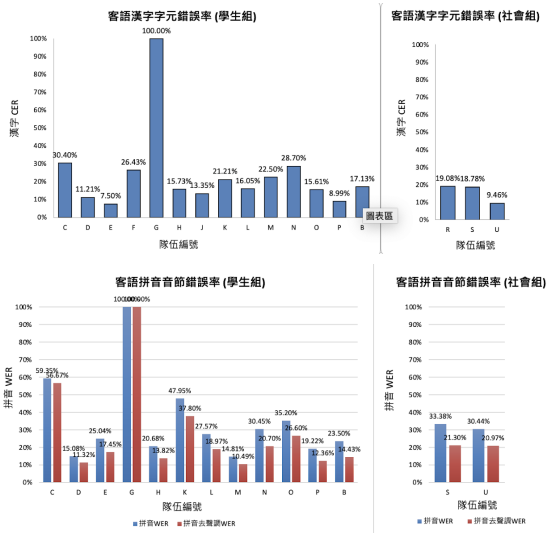

Figure 2: Challenge 2025 final-test result charts.

2025.

We provide end-to-end scripts for data preparation, training, inference, and evaluation. Minimal examples and notes:

Notes. Track 1 defaults keep asterisks and punctuation unless `--strip_asterisk`/`--strip_punct` is specified. Track 2 drops starred syllables by default and supports optional split-tone merging with `--fix_split_tone`. All runs use seed 1337.

Track 1:

```
python prepare_hakka_track1.py --root
    HAT-Vol2 \
  --drop_mispronounce --relative_audio_path
python train_whisper_lora_track1.py \
  --train_jsonl
      HAT-Vol2/manifests_track1/train.jsonl \
  --dev_jsonl
      HAT-Vol2/manifests_track1/dev.jsonl
python infer_track1.py --eval_root
    FSR-2025-Hakka-evaluation \
  --outfile predictions_track1.csv --model
      openai/whisper-large-v2 \
  --lora_dir runs/track1/lora_v2_r16_e3
python eval_track1_cer.py --key_dir
    FSR-2025-Hakka-evaluation-key \
  --hyp predictions_track1.csv
```

Alternative (recommended for Track 2 with v3-turbo):

```
python train_whisper_lora_track2.py
    --base_model
    openai/whisper-large-v3-turbo --out_dir
    exp_track2_whisper_large_v3_turbo_lora
python infer_track2.py --eval_root
    FSR-2025-Hakka-evaluation \
  --outfile predictions_track2.csv --model
      openai/whisper-large-v3-turbo \
  --lora_dir
      exp_track2_whisper_large_v3_turbo_lora
```

Track 2:

```
python prepare_hakka_track2.py --data_root
    HAT-Vol2 \
```

```
  --out_dir HAT-Vol2/manifests_track2
      --exclude_mispronounced
python train_whisper_lora_track2.py
    --base_model
    openai/whisper-large-v3-turbo --out_dir
    exp_track2_whisper_large_v3_turbo_lora
python infer_track2.py --eval_root
    FSR-2025-Hakka-evaluation \
  --outfile predictions_track2.csv --model
      openai/whisper-large-v2 \
  --lora_dir exp_track2_whisper_large_lora
python eval_track2_ser.py --key_dir
    FSR-2025-Hakka-evaluation-key \
  --hyp predictions_track2.csv
```

8 Error Analysis

Common errors include character/phonetic substitutions and occasional short repeats; we monitor n-gram repetition to detect degeneration. Performance degrades mildly for longer utterances; bucketed analysis suggests length-aware decoding or better segmenting could help.

Examples. Sampled warm-up mismatches:
(003jh5p8hd.wav) ref: 大家攏無仰子**啲**隨拹你人救出去; hyp: 大家攏無仰子**項**隨拹你研究出去.
(03qw9gfad7.wav) ref: 食**著**幾隻草蜢**乜好啊**; hyp: 食**到佢**隻草蜢**毋會好啊**.
(04qied7gz8.wav) ref: …這兜**地動無幾著呢**莊頭…; hyp: …這兜**地圖無幾臭呢啊**莊頭…

These illustrate homophone/near-neighbor substitutions and local phrase alterations; stronger language modeling or constrained decoding may mitigate such errors. For Pinyin (Track 2), common patterns include tone-digit confusions and occasional effects from star-syllable handling; our normalization reduces such artifacts.

9 Conclusion

We provide a concise, reproducible baseline for both tracks of FSR-2025 Hakka ASR using Whisper+LoRA. Future work includes dialect-aware adaptation, LM-rescoring for Hanzi, refined Pinyin normalization, and temperature/beam tuning.

Limitations

Our results are based on the provided HAT-Vol2 training data and the official warm-up set. We do not explore external language models or data augmentation; Pinyin normalization choices (e.g., starred syllables) can affect SER.

Acknowledgments

We thank the FSR-2025 organizers and dataset providers.

References

FSR2025. 2025. Formosa speech recognition challenge 2025: Hakka asr. Challenge. Warm-up evaluation set and official task description.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Association for Computational Linguistics.

HAT-Vol2. 2024. Hat-vol2: Taiwanese hakka speech corpus. Dataset. ~60 hours; Dapu and Zhao'an dialects; 16 kHz mono.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.