Advancing Thai Speech-to-Text: Deep Dive into Fine-Tuning Thonburian Whisper

Whitepaper by:

Engineering & Research Team - Tollring

Wachiravit Modecrua, Director of AI Labs - Amity solutions

Content

1. Introduction

Voice is becoming an increasingly vital interface in digital experiences, from customer support to virtual assistants. But building accurate speech-to-text (STT) systems for non-English languages is still one of AI's toughest challenges.In particular, Thai presents a unique set of linguistic hurdles: it is tonal, lacks spaces between words, and is frequently spoken with informal phrasing, regional accents, or interspersed English terms. These complexities create substantial barriers for standard STT models, which are typically trained on English or space-delimited languages.

In this whitepaper, we present our efforts to overcome these challenges by fine-tuning large-scale speech recognition models, specifically, Whisper, for Thai STT in real-world settings. Our goal is to create accurate, production-grade models that serve the practical demands of customer-facing industries such as call centers and insurance providers, where transcription accuracy directly impacts service quality and automation potential.

We document our full technical approach, from curated dataset design and advanced fine-tuning strategies (including Full, LoRA, and hybrid methods) to diarization (i.e., the process of automatically identifying and segmenting an audio recording based on the identity of the speaker) performance and downstream improvements with LLM-based post-processing. Through rigorous experimentation and benchmarking, we demonstrate not only substantial gains in Word Error Rate (WER) but also operational benefits such as cost savings, real-time inference capability, and adaptability across speech domains.

This work moves us closer to scalable, reliable Thai-language voice AI that functions well for Thai speakers, bringing higher accuracy, scalability, and reliability to Thai-language speech applications at scale.

2. Dataset Construction and Properties

Our work began with data. Recognizing that the performance of an STT model depends heavily on dataset design and diversity, we assembled and partitioned multiple high-quality, domain-relevant Thai audio/text datasets.

- **Contact Center Dataset**:
  - 615 segments, random 80%/20% train-test split
- **Insurance Dataset**:
  - 422 segments, even 50%-50% split

We carefully built the dataset to include a balanced representation of speakers and acoustic environments, allowing for a more nuanced assessment of fine-tuning regimes.

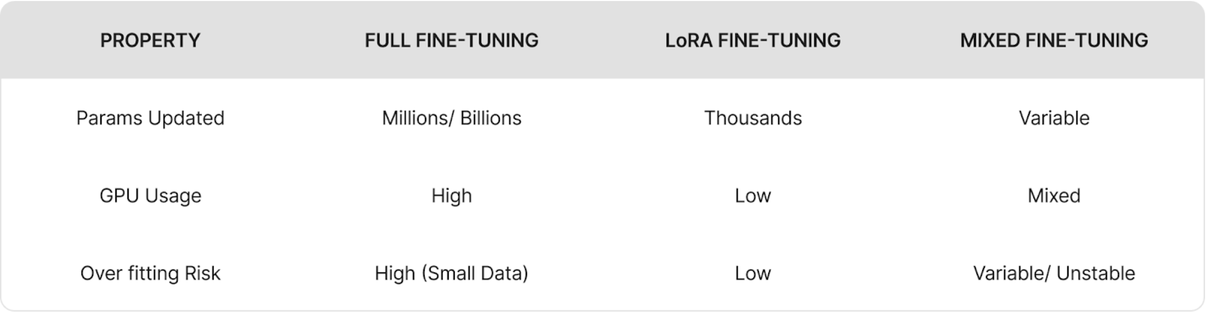
3. Fine-Tuning Methodologies

To truly push the limits of Whisper on Thai speech, we evaluated several distinct fine-tuning methodologies:

1. **Full Fine-tuning**
   Full fine-tuning updates every parameter in the model during training, allowing adaptation at every layer.
   - **Parameter Count:** All model weights (millions/billions)
   - **Pros:** Highest flexibility, potential for strong fit to the dataset
   - **Cons:** High GPU usage, longer training/inference times, risk of overfitting, especially with limited data

2. **LoRA (Low-Rank Adaptation) Fine-tuning**
   LoRA fine-tuning is parameter-efficient:
   - **How it works:** Freezes nearly all model weights, updating only small projection submodules (e.g., "q_proj", "v_proj").
   - **Parameter Count:** Few thousand
   - **Pros:** Rapid adaptation, very low GPU cost, robust against overfitting, enables rapid experimentation
   - **Cons:** Less expressive for large distribution shifts

3. **Mixed/ Chained Fine-tuning**
   To exploit the strengths of both, we explored **mixed approaches**, where the model is first LoRA-tuned then full-finetuned on additional data, or vice versa.
   - **Intent**: Hybridize rapid adaptation and model capacity
   - **Caveat**: Nonlinear/unstable convergence in some cases



PROPERTY	FULL FINE-TUNING	LoRA FINE-TUNING	MIXED FINE-TUNING
Params Updated	Millions/ Billions	Thousands	Variable
GPU Usage	High	Low	Mixed
Over fitting Risk	High (Small Data)	Low	Variable/ Unstable

Table 1: Fine-tuning Methodologies

4. Experimental Results

Performance by Dataset & Method

Word Error Rate (WER) is a standard evaluation metric used to measure the accuracy of speech recognition systems, particularly in automatic speech recognition (ASR) tasks.

WER calculates the difference between a predicted transcript and a reference (correct) transcript, based on how many word-level edits are needed to turn the predicted version into the correct one. **Lower WER means better transcription accuracy.**

Results

- **Typhoon2 Baseline:**
  - ContactCenter Dataset: WER 0.90
  - Insurance Dataset: WER 0.65

- **Thonburian Baseline:**
  - ContactCenter Dataset: WER 0.53
  - Insurance Dataset: WER 0.33

- **Local Cloud Speech To Text (STT) Service Baseline:**
  - ContactCenter Dataset: WER 0.47
  - Insurance Dataset: WER 0.29

- **Amity - Full Finetuning:**
  Larger datasets yielded substantial WER reductions.
  - ContactCenter Dataset: WER 0.41
  - Insurance Dataset: WER 0.21

- **Amity - LoRA Finetuning:**
  When the dataset is limited, LoRA adaptations outperformed the baseline consistently.
  - ContactCenter Dataset: WER 0.41
  - Insurance Dataset: WER 0.25

- **Amity - Mixed Finetuning**:
  - ContactCenter Dataset: WER 0.36
  - Insurance Dataset: WER 0.30

WER Performance on Thai Datasets (Lower is Better)

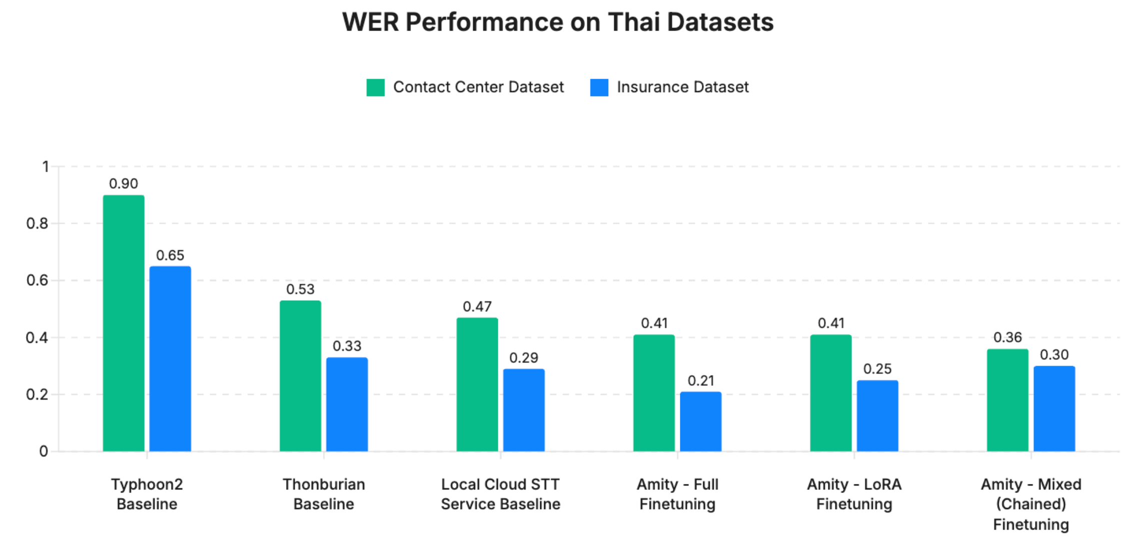


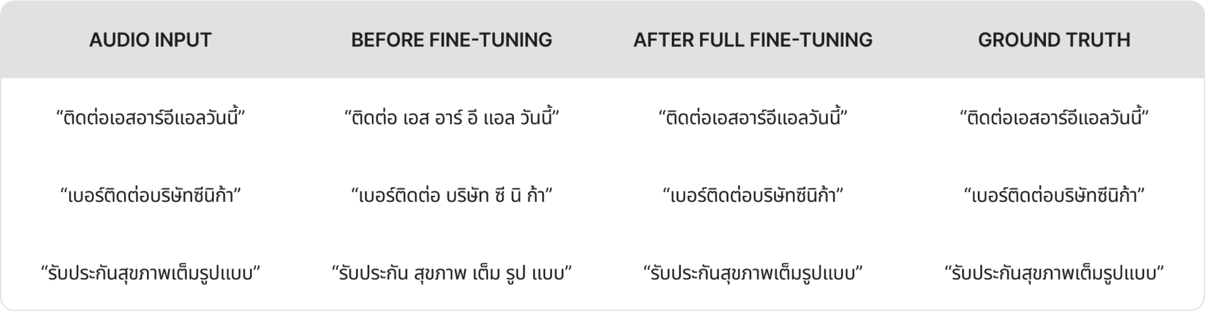
Figure 1: WER Comparison Between Amity Fine-Tuning Methods and Other Service Baselines

Observations:

- **Scaling Effect**: More (high-quality) speech data → better model generalization and performance with full finetuning.
- **LoRA Advantage**: For data-limited scenarios, LoRA adapts quickly, uses less GPU/memory, and is resilient to overfitting—ideal for rapid development cycles.
- **Hybrid Instability**: Chaining LoRA and full finetuning produced inconsistent results, hinting at complex interactions between model adaptation stages, possibly due to catastrophic forgetting or conflicting parameter updates.

Qualitative Examples: Before and After Fine-tuning

To illustrate the impact of our fine-tuning strategies, we present real output examples from Thonburian Whisper on the Insurance dataset:



AUDIO INPUT	BEFORE FINE-TUNING	AFTER FULL FINE-TUNING	GROUND TRUTH
“ติดต่อเอสอาร์อีแอลวันนี้”	“ติดต่อ เอส อาร์ อี แอล วันนี้”	“ติดต่อเอสอาร์อีแอลวันนี้”	“ติดต่อเอสอาร์อีแอลวันนี้”
“เบอร์ติดต่อบริษัทซีนิก้า”	“เบอร์ติดต่อ บริษัท ซี นิ ก้า”	“เบอร์ติดต่อบริษัทซีนิก้า”	“เบอร์ติดต่อบริษัทซีนิก้า”
“รับประกันสุขภาพเต็มรูปแบบ”	“รับประกัน สุขภาพ เต็ม รูป แบบ”	“รับประกันสุขภาพเต็มรูปแบบ”	“รับประกันสุขภาพเต็มรูปแบบ”

Table 2: Example Output Improvements from Fine-Tuning on the Insurance Dataset

Observation:

- The model pre-finetuning splits Thai brand names and compounds erroneously and inserts spaces, reflecting its unfamiliarity with language-specific (especially branded) terms.
- After fine-tuning (with either full or LoRA adaptation), the model correctly transcribes and composes brand names and domain terminology with greater consistency.

Inference Strategy

Our production pipeline supports:

- **Language-code inference** (provides best results for shorter segments, but can degrade longer-segment accuracy)

- **Prompting**: By passing transcript references as context, up to 90% segments matched manual transcriptions, at the cost of roughly 2x slower inference speeds.

5. Speaker Diarization: Applying STT Model Adaptation Techniques

Recognizing the need for diarization in bilingual/multispeaker Thai recordings, we applied similar training discipline:

- Segmented datasets into Smaller (12) units and Larger (615) units, 5 seconds per segment.
- Utilized full-finetuning and clustering via Diarizer library.
- DER **dropped** from 31.3% (Raw Insurance Data) to 8.13% (fine-tuned), validating the transferability of structured fine-tuning strategy.

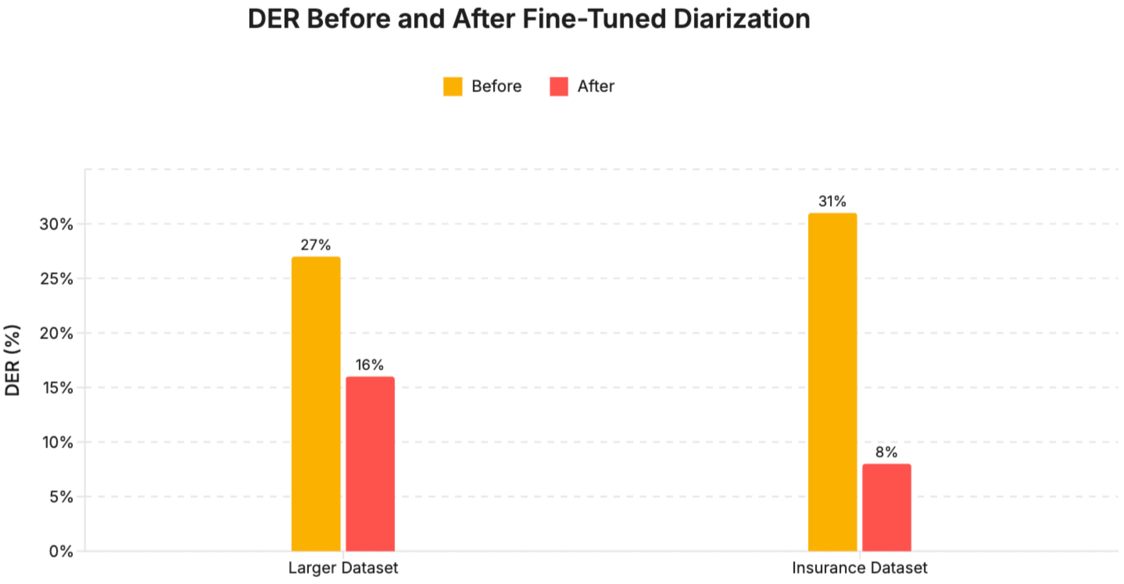
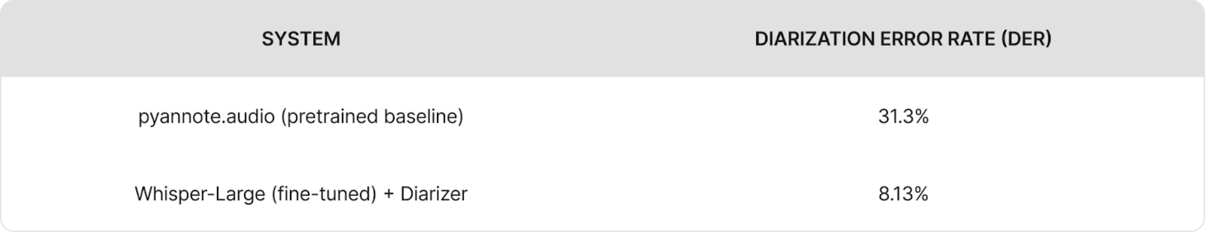


Figure 2: DER before and after Fine-tuned Diarization

Baseline Model Performance

Our initial baseline for speaker diarization was the widely used pyannote.audio pipeline, which relies on pretrained speaker embeddings and spectral clustering. On the Insurance dataset, this vanilla approach, without any Thai-specific adaptation, produced a Diarization Error Rate (DER) of 31.3%, with principal errors in distinguishing overlapping and code-switched speakers.

Building on this, we used our fine-tuned Thonburian Whisper-Large model as the front-end for audio segmentation, and combined it with the Diarizer library for speaker clustering. This yielded a marked improvement, dropping DER to 8.13% on the same evaluation set.



SYSTEM	DIARIZATION ERROR RATE (DER)
pyannote.audio (pretrained baseline)	31.3%
Whisper-Large (fine-tuned) + Diarizer	8.13%

Table 3: DER Comparison between Baseline vs. Fine-Tuned Diarization

6. Cost and Operational Analysis

An effective production STT must balance accuracy, latency, and cost:

- **Model/Server**:
  - Largest model: Whisper (1.6B parameters)
  - GPU: RTX A3090, 24GB, $0.37/hr (at cloud rates)
- **Throughput**:
  - 75 samples, average duration 5.47s, total ~410s audio processed in ~97s (throughput 4.2x realtime)
- **Cost Estimation**:
  - Ours: $0.0061/minute audio
  - Local Cloud Speech To Text Service Baseline: $0.01/minute audio

This approach yields **substantial cost savings** while improving quality.

Model & Deployment Specs

- **Production Model:** Whisper-Large (1.5B params, 1.6B w/ adapters), RTX A3090 24GB GPU
- **Batch Size:** 8, optimized kernel, ~75 utterances per batch
- **Throughput:** ~4.2x real-time (tested on 410s audio → 97s processing)

Normalization Method:

- Measured from GPU time per batch, inclusive of disk I/O, no batching latency.
- For 1 hour audio:
  - Ours: 1 hr audio / 4.2x real-time = 14.3 GPU mins → $0.37 (A3090 instance per hour)

Key Insights:

- Our solution runs at **~39% lower cost per hour**, plus added benefit of in-house control, model customization, and no data privacy risks.
- Scaling up to 1000+ hours/month: projected monthly savings >$230.

7. Future Improvement and LLM-Assisted Autocorrection

Initial LLM Experiments: Brand Name Correction

To handle persistent transcription challenges around rare/ambiguous Thai brand terms, we built a post-processing pipeline:

- Method: Output from the fine-tuned Whisper is passed to a lightweight LLM (e.g., distilled GPT2-Thai, tuned on brand lexicons).
- Pilot batch: Insurance test set, 20 samples with ambiguous brand terms.
- Performance:
  - Pre-LLM Correction: 78% accuracy (manual ground-truth)
  - Post-LLM Correction: 93% accuracy; majority of critical “brand” fixes recovered
- Transformation logic: Contextual matching and spelling similarity help the LLM correct syllabic or phonetic errors in transcriptions, particularly for OOV Thai brand terms.

Roadmap:

- Expand LLM with slot-based correction for more insurance/product terms
- Integrate into main pipeline and evaluate latency/cost at inference scale
- Explore auto-correction not only for brands, but common colloquial/idiomatic Thai

8. Conclusions & Roadmap

By leveraging diverse data, multi-pronged fine-tuning (full/LoRA/mixed), and robust deployment, we've achieved:

- State-of-the-art Thai STT results, consistent WER reductions vs commercial and open baselines
- Dramatic improvements in speaker diarization
- Significant reduction in serving costs
- Flexible, production-grade architectures for scalable, reliable inference

Next steps:

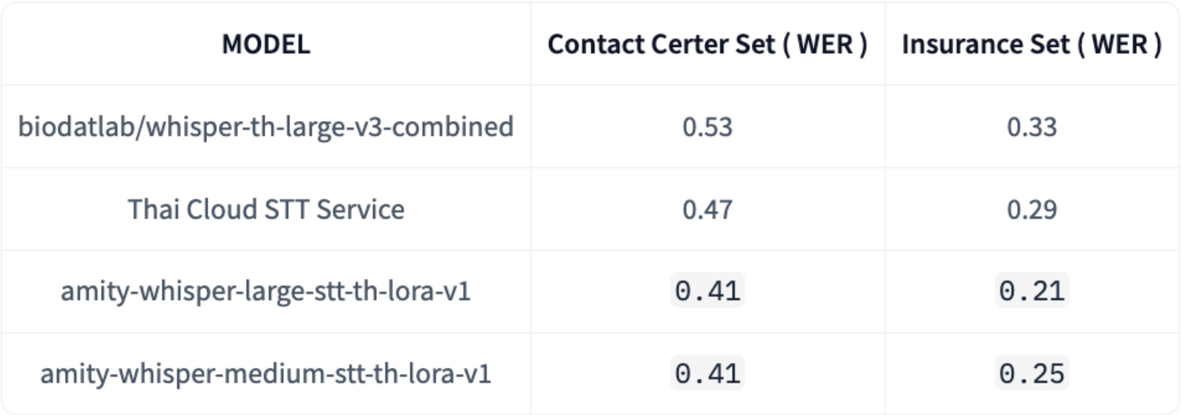
- Expanding dataset multilinguality
- Exploring more stable hybrid-adaptation methods
- Integrating on-the-fly speaker adaptation

Appendix A: Amity Thai Speech-to-Text Models Hosted on Hugging Face

As part of our commitment to open research and the advancement of Thai speech technology, we have published two Amity Speech-to-Text (STT) models on Hugging Face, fine-tuned from OpenAI's Whisper architecture using LoRA for efficient adaptation to Thai. Both models are optimized for high accuracy in transcribing Thai speech, including conversational and domain-specific audio.

The available models are:

1. **Amity Whisper Large LoRA V1 (Thai)**
    - https://huggingface.co/amityco/amity-whisper-large-stt-th-lora-v1
    - Designed for maximum transcription accuracy, suitable for research or production environments where precision is critical.
    - This model is a fine-tuned version of biodatlab/whisper-th-large-v3-combined on private enterprise call-center voice datasets. It achieves the following results on the real industry (contact center, insurance) test set
2. **Amity Whisper Medium LoRA V1 (Thai)**
    - https://huggingface.co/amityco/amity-whisper-medium-stt-th-lora-v1
    - A lighter-weight model offering a balance between speed and accuracy, ideal for applications requiring lower latency or reduced compute resources.
    - This model is a fine-tuned version of biodatlab/whisper-th-medium-combined on private enterprise call-center voice datasets. It achieves the following results on the real industry (contact center, insurance) test set



MODEL	Contact Certer Set (WER)	Insurance Set (WER)
biodatlab/whisper-th-large-v3-combined	0.53	0.33
Thai Cloud STT Service	0.47	0.29
amity-whisper-large-stt-th-lora-v1	`0.41`	`0.21`
amity-whisper-medium-stt-th-lora-v1	`0.41`	`0.25`

Table 4: WER comparison of four speech-to-text models across Contact Center and Insurance datasets

Each repository includes pretrained weights, inference scripts, and usage guidelines, enabling developers and researchers to quickly integrate these models into their workflows. By making both variants publicly available, we aim to provide flexibility for different computational and accuracy requirements, and to encourage further experimentation and innovation in Thai STT systems.

Appendix B: Amity Voice Segmentation Model Hosted on Hugging Face

To support downstream speech-processing tasks such as speaker diarization, transcription, and call analytics, we have released the **Amity Voice Segmentation Model**: https://huggingface.co/amityco/amity-voice-segmentation-01 on Hugging Face.

This model is designed to accurately segment continuous audio streams into distinct speech segments, marking the start and end times of spoken content while ignoring silence, background noise, and non-speech events. It is trained on domain-relevant Thai audio data and optimized for performance in real-world conditions such as telephone calls and conversational recordings.

This model is a fine-tuned version of pyannote/segmentation-3.0 on the `amityco/sample-voice-12-records` and private datasets from Real Contact-Center Data. It achieves the following results on the evaluation set:

- Loss: 0.1038
- Model Preparation Time: 0.0025
- Der: 0.0468
- False Alarm: 0.0280
- Missed Detection: 0.0188
- Confusion: 0.0

By making the model publicly available, we aim to provide researchers, developers, and industry practitioners with a ready-to-use tool for building robust Thai voice-based applications. The Hugging Face repository includes pretrained weights, sample inference scripts, and integration guidelines to help accelerate adoption and experimentation.

Evaluation Result

Our initial baseline for speaker diarization was the widely used `pyannote.audio` pipeline, leveraging pretrained speaker embeddings and spectral clustering. On the Insurance dataset, this vanilla approach—without any Thai-specific adaptation—produced a Diarization Error Rate (DER) of 31.3%, with principal errors in distinguishing overlapping and code-switched speakers.

Building on this baseline, our fine-tuned approach paired the **fine-tuned Thonburian Whisper-Large** model as the acoustic front end for segmentation with the **Diarizer** clustering library. This yielded a marked improvement, reducing the DER to 8.13% on the same evaluation set (see *Table 3: DER Comparison between Baseline vs. Fine-Tuned Diarization, p.6*).