End-to-End Speech Translation for Low-Resource Code-Switching

System Design and Model Evaluation

La Quang Chien
Tran Thi Vy Anh
Vo Huyen Khanh May

December 16, 2025

National Economics University
Hanoi, Vietnam

Outline

Introduction

Methodology I

Methodology II

Results

Analysis

Conclusion

Introduction

Motivation and Problem Definition

The Linguistic Landscape

- Vietnamese-English Code-Switching (CS) is ubiquitous in modern Vietnam, appearing frequently in media and professional settings.
- This presents a specific challenge: reconciling the tonal precision of Vietnamese with the stress-timed nature of English within a single acoustic stream.

The Data Gap

- There is a critical shortage of high-quality, timestamped CS data.
- Existing corpora are predominantly synthetic or insufficient in volume for deep learning applications.

Architectural Fragility

- Traditional cascaded systems (ASR $\rightarrow$ MT) amplify errors downstream.
- End-to-End (E2E) models require massive data to converge, which is unavailable for this specific language pair.

Project Objectives

This project targets two distinct domains:

1. Engineering Objective

- Construct a scalable, fault-tolerant software pipeline to synthesize a dataset from raw YouTube audio.
- Orchestrate ingestion, AI-assisted pre-labeling, and distributed human verification into a seamless workflow.
- A glorious way to say: Create a manual data labelling pipeline, from data crawling to data annotating.

2. Scientific Objective

- Benchmark a E2E architecture against a Whisper baseline.
- Evaluate if a custom encoder-decoder arrangement can outperform massive pre-trained models in low-resource CS settings.

Methodology I

Data ...

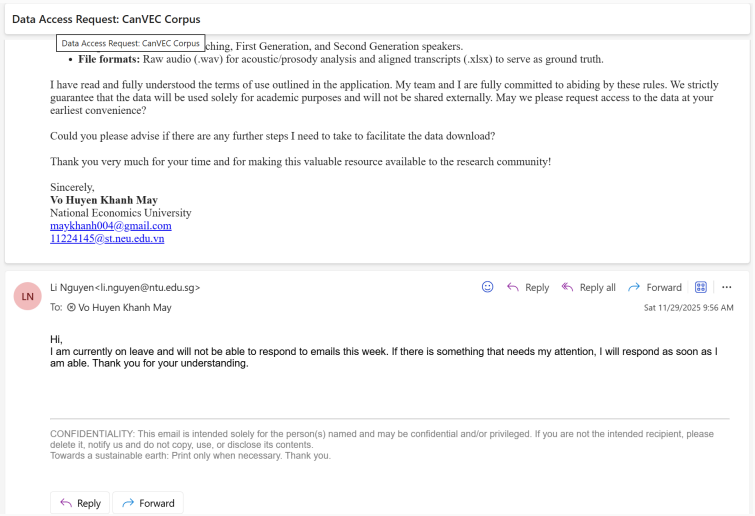


Figure 1: không có data thì mình tự làm ...

The Illusion of Free Choice

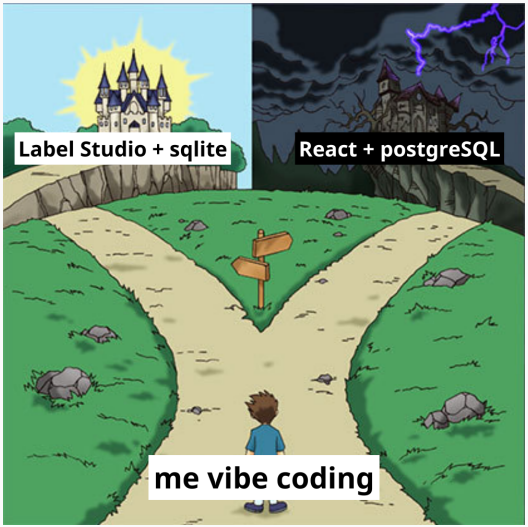


Figure 2: I chose the dark path, thanks Claude

System Design

Tech Stack & Persistence

- **PostgreSQL:** Chosen for ACID compliance to manage complex relationships between channels, videos, chunks, and segments.
- **FastAPI & SQLModel:** Acts as the single source of truth, validating data types at the API boundary to reject malformed inputs.

Ingestion & Pre-labeling

- **Standardization:** ‘yt-dlp‘ normalizes audio to 16kHz mono AAC. ffmpeg cut audio into 5-minute-and-5-second chunks, ready to be fed into the model.
- **AI Pipeline:** Gemini 2.5 utilized with a "Senior Linguistic Data Specialist" persona to ensure tonality preservation and millisecond timestamp precision.

Verification Workbench

- Custom React frontend with waveform visualization (very cool).
- Implements "Ghost Locking" to manage distributed concurrency, preventing overwrite collisions among annotators.

Engineering Challenges and Solutions

1. Overlap Management

- *Problem:* 5-second safety buffers at chunk boundaries created duplicate data points.
- *Solution:* A deterministic SQL heuristic. Any segment starting $\geq$ 300s relative time is strictly dropped from the export $\rightarrow$ Ensures zero duplication without complex fuzzy merging.

2. API Rate Limiting

- *Problem:* Frequent 'RESOURCE_EXHAUSTED' (HTTP 429) errors from the Gemini API during bulk ingestion.
- *Solution:* Use 32 API keys. A resilient state machine that treats API quotas as a pooled resource and **pray**. Quota for 2.5 Flash is 1M requests per day we didn't even cross that but still get quotaed.
- The system cycles keys, escalates from Flash to Pro tiers automatically, and enters a global sleep state only when all resources are depleted.

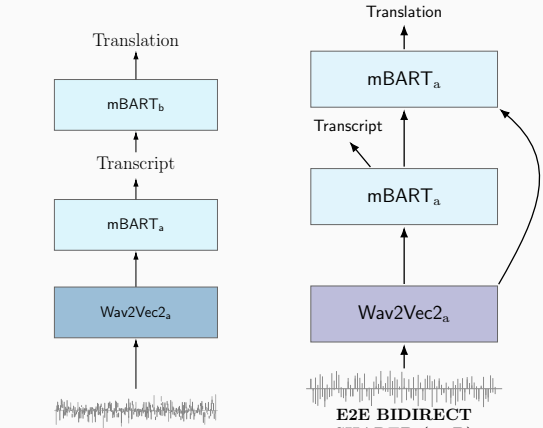
UI Demo Time :)

Pov: You are manually going through these audios and transcripts to traumatize yourself.

Methodology II

Intended Model Architecture Comparison

Originally, we aimed to contrast an cascade approach (using two seperate decoder for each task) against an end-to-end approach (using one decoder for both tasks).



Actual Comparison Setup

However, we did not have enough time (and later, computational resource), so we change to Whisper (for latest, unified ASR+MT architect) vs. E2E (for traditional encoder/decoder approach)

Feature	Baseline (Whisper)	Experimental (E2E)
Core Model	`whisper-medium`	Wav2Vec2 + Adapter + mBART-50
Encoder	Log-mel Spectrogram Transformer	`wav2vec2-large-xlsr-53`; unfreeze after 1 epoch
Decoder	Text Decoder (English-biased)	`mbart-large-50`
Input	80-band Mel Spectrogram	Raw Waveform (16kHz)
Strategy	Multitask with `forced_decoder_ids`	Direct Speech-to-Text Translation

Table 1: Architectural Comparison

Experimental Setup: Data Specification and Metrics

Dataset Specification

- **Volume:** 63.48 hours of human-verified YouTube CS audio.
- **Sanitization:** Removal of non-breaking spaces, markdown artifacts, and heuristic filtering ($0.5s < \text{duration} < 30s$).
- **Splitting Strategy:** Data isolation enforced at the **Video Level** (80/10/10 split).
- This prevents acoustic leakage (speaker identity, background noise) between train and test sets.

Evaluation Metrics

- **ASR:** Word Error Rate (WER) and Character Error Rate (CER). CER is prioritized due to the ambiguity of Vietnamese word boundaries.
- **Translation:** BLEU (n-gram overlap) and chrF (Character n-gram F-score), utilizing ‘sacrebleu‘.

Experimental Setup: Hardware and Configurations

To accommodate resource constraints, we utilized distinct configurations for development (debugging) and production (convergence).

Parameter	Development (RTX 3060)	Production (H100 SXM)
VRAM	12 GB	80 GB
Whisper Variant	`whisper-tiny` (39M)	`whisper-small` (244M)
Batch Size	1	32 (Whisper) / 1 (E2E)
Grad Accumulation	8 Steps	2 Steps (Whisper) / 32 (E2E)
Effective Batch Size	16	64
Precision	FP16 (Mixed)	BF16 (Brain Float 16)
Optimization	Full Encoder Frozen	Feature Encoder Frozen

Table 2: Parameter Configuration Comparison

Experimental Setup: Intended vs. Actual Parameter Configurations

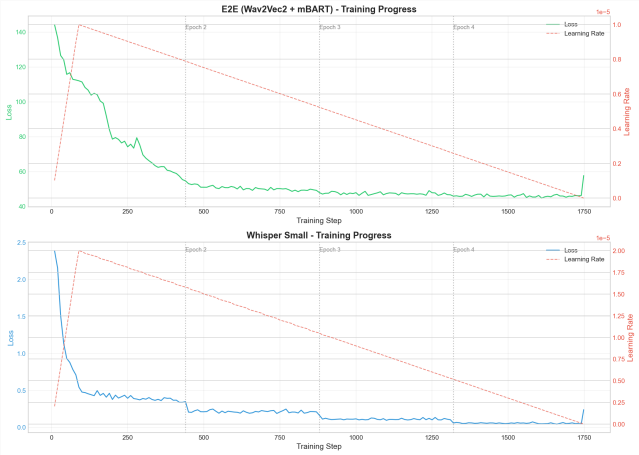
Who would have thought 80GB VRAM is not enough, not us though. Also we realized Whisper doesn't support Vietnamese translation so good luck fine-tuning with these limited data.

	E2E Architecture		**Whisper Baseline**	
Parameter	*Original (Intended)*	**Optimized (Actual)**	*Original (Intended)*	**Optimized (Actual)**
Model/Encoder	`wav2vec2-large-xlsr-53`	`wav2vec2-base`	`whisper-medium`	`whisper-small`
Param Count	∼317M (Encoder Only)	**94M** (Encoder Only)	769M	**244M**
Task Scope	Multitask (ASR + ST)	Multitask (ASR + ST)	Multitask (ASR + ST)	**ASR Only**
Freezing Strategy	Unfreeze after Epoch 1	**Fully Frozen Encoder**	Full Finetuning	Full Finetuning
Batch Size	8	**2**	24	**32**
Grad Accumulation	4 steps	**16 steps**	2 steps	**1 step**
Effective Batch Size	32	**64**	48	**64**
Learning Rate	5.0×10^{-6}	1.0×10^{-5}	1.0×10^{-5}	2.0×10^{-5}
Epochs	3	**4**	3	**4**

Results

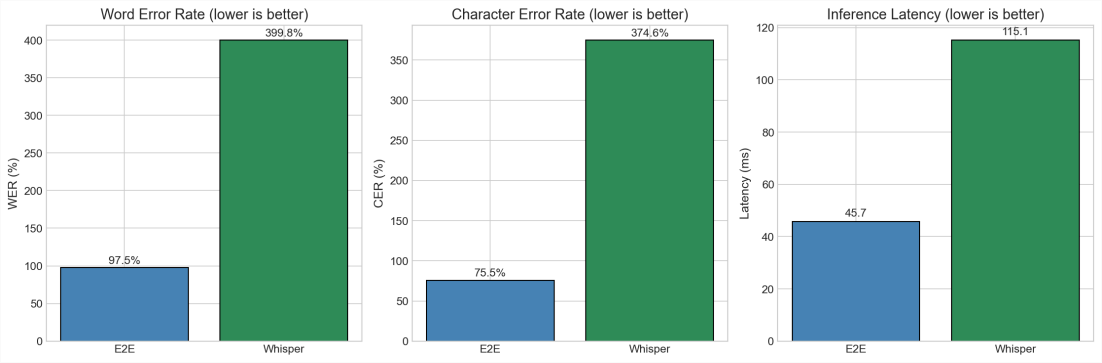
Training Dynamics

Whisper's loss drops precipitously, suggesting overfitting to a looping pattern, while E2E loss plateaus due to the frozen encoder.



Results: ASR Performance and Latency

The divergence from targets is extreme. Whisper suffered from massive insertion errors (looping), while the E2E model stalled at near 100% error rates.



Results: Translation Quality

Despite the theoretical capability for simultaneous translation, the E2E model failed to capture semantic signal, resulting in near-zero BLEU scores.

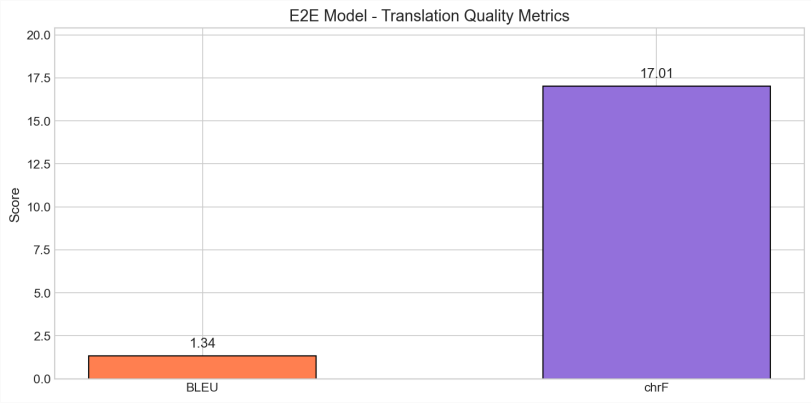


Figure 4: BLEU and chrF scores indicate a failure to map English-heavy inputs to Vietnamese outputs.

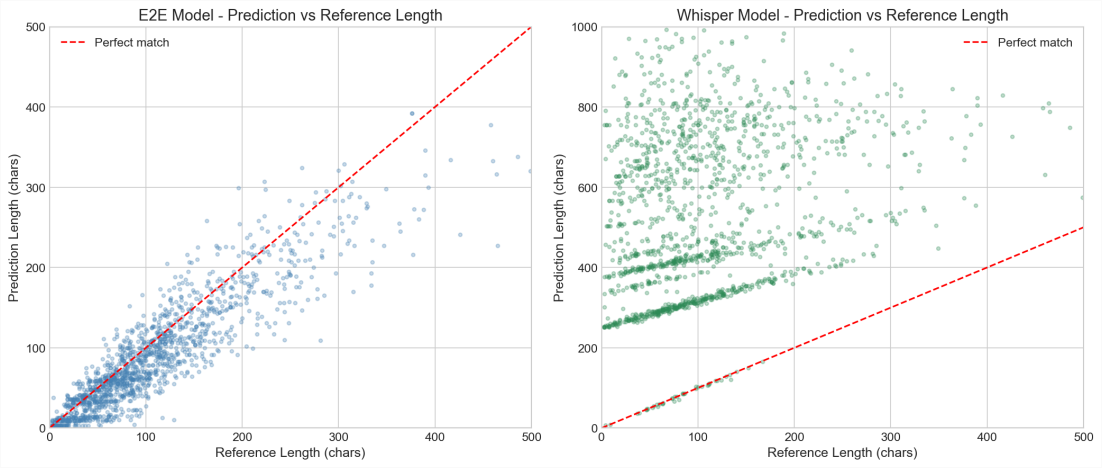
Raw Prediction Showcase

Let's see what the model predicted!

Analysis

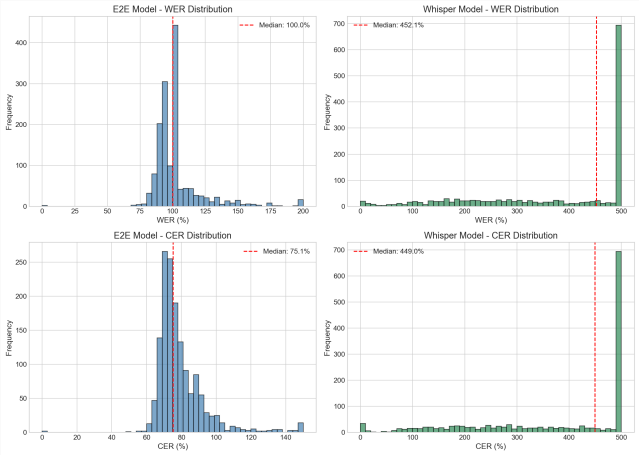
Failure Analysis: Hyper-graphia vs. Aphasia

The length analysis confirms that Whisper (Green) loops infinitely (Hyper-graphia), while E2E (Blue) under-predicts or outputs silence (Aphasia).



Failure Analysis: Error Distribution

The distribution of error rates further highlights the dichotomy. Whisper's median WER is >450% due to insertion penalties, while E2E clusters strictly at 100% (total failure).



Root Cause

We identified two fundamental architectural conflicts that caused these failure modes.

1. Whisper: Multitask Conflict

- The <|translate|> token in Whisper is strictly trained for **Any-to-English**.
- The model were forced to output Vietnamese while using the translation token.
- **Result:** The model faced opposing objectives (Token says "English", Weights say "English", Labels say "Vietnamese"), causing attention collapse and looping.

2. E2E: Frozen Encoder cause Domain Mismatch

- To fit 80GB VRAM, we used `wav2vec2-base` and **froze** the encoder entirely (previously intended to unfreeze after 1 epoch).
- This encoder was pre-trained **exclusively on English** (LibriSpeech).
- **Result:** Without fine-tuning, the English acoustic priors could not map to Vietnamese tonal phonemes. The shallow adapter layers were insufficient to bridge this deep acoustic domain gap.

Conclusion

Conclusion

Engineering Success :)

- The **Data Factory** proved to be a robust, scalable scientific instrument.
- We successfully operationalized the ingestion and verification of code-switched audio, solving distributed concurrency and data integrity at (a small) scale.

Scientific Failure :(

- Architectural elegance cannot compensate for data scarcity.
- The complex multitask objective requires a density of training signal that the current 60-hour dataset cannot provide.
- Freezing English-centric encoders is detrimental for tonal language tasks in low-resource settings.

Future Work that will never got done

1. Scaling the Signal

- Leverage the Data Factory to scale the corpus from 60 to **500 hours**.
- Ingest a broader variance of audio sources (vlogs to podcasts) to stabilize acoustic priors.

2. Decomposing the Objective

- Temporarily decouple ASR and Translation tasks.
- Train separate baselines to convergence before fusing them into a joint E2E system.
- That said, we would want to rent 2 or 4 H100 to fully finetune the model (had we got the money)

3. Curriculum Learning

- Implement a warm-up schedule for Whisper.
- Train on short, high-confidence segments first to stabilize the attention mechanism before introducing complex overlapping code-switching.