EchoLang
Project Report

Real-time speech processing
and multilingual translation tool

Dhruv Maheshwari

Table of Contents

10/08/2025
Thursday

PROJECT REPORT - EchoLang

~ Dhruv Maheshwari

Problem Statement:

Real time speech to text (STT) translation and transcription tool, focused on a mix of Hindi/Tamil with English. (Transcription and translation of mixed Hindi-English (Hinglish) and Tamil-English speech into English text.)

- Considering use cases:
    a. Accessing a Yellow Page Directory for blue collar workers living in Tier 2/3 cities.
    b. Automated creation of medical prescription through passive observation of a Doctor-Patient conversation focused on people living in Tier 2/3 cities.

Introduction:

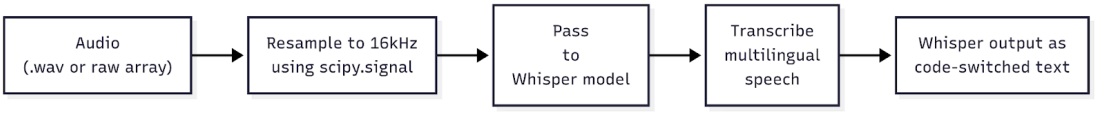
This report details the 5 week development process of the project, and final product - EchoLang as an outcome. The focus of this project was to create a multilingual, code-switched chatbot system intended to help people living in Tier 2/3 Indian cities access digital services like Yellow Page Directories. The system is designed to translate mixed Hindi-English and Tamil-English inputs into actionable English queries while handling real-time speech-to-text transcription, where underserved communities are enabled to access healthcare, emergency, and maintenance services through natural language by removing barriers to digital literacy and allowing audio/text input in regional languages.

The bulk of the project depends on the pipeline using OpenAI's Whisper for Automatic Speech Recognition (ASR) and Meta's NLLB-200 for Machine Translation (MT), alongside Sentence-BERT embeddings for semantic classification. Whisper expertly handles mixed Indian languages with English and acoustic conditions with limited resources. After transcription, the output is run through Meta's NLLB-200, which detects and converts regional language fragments into standard English, creating a normalized and semantically coherent text that can be processed further. Furthermore, the output from NLLB-200 is passed on to Sentence-BERT for semantic classification, which interprets free-form user queries by generating fixed-size vector embeddings of each input sentence. These embeddings are then compared against precomputed embeddings of defined service categories using cosine similarity. This allows the system to semantically match user intent, even when the phrasing does not directly align with predefined keywords.

Core Features

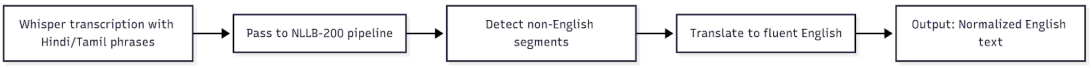
1. **Automatic Speech Recognition** (ASR):
   Tool Used: OpenAI Whisper

   The Speech-to-text (STT) feature of this model is capable of transcribing multilingual and code-switched speech in real time. It supports Hindi-Tamil-English combinations, detects spoken segments, and produces transcriptions even under low-resource or noisy conditions. The system includes a resampling function that converts raw audio from 32kHz to 16kHz using signal interpolation with SciPy, ensuring compatibility with Whisper's expected input format. The inclusion of resampling allows the system to support a broader range of audio inputs at runtime.



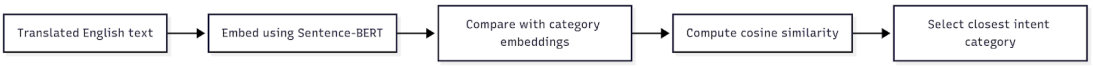
2. **Machine Translation** (MT):
   Tool Used: Meta NLLB-200

   This process is in charge of translating regional language fragments, like those in Tamil or Hindi, into fluent English after the user's speech has been converted to text. In order to clean up Whisper's mixed outputs, where regional phrases might appear in code switched format, this step is essential. NLLB-200 can preserve important contextual elements like named entities and domain-specific terminology because it was specifically trained on low-resource Indic languages. The downstream classifier's inputs are ideally semantically correct and structurally consistent by the model. Because of its broad coverage of the Indian languages and effective inference profile, this makes it ideal for prototyping in limited settings as will be found in effect.



3. **Intent Classification and Semantic Understanding**:
   Tool Used: Sentence BERT (all-MiniLM-L6-v2)

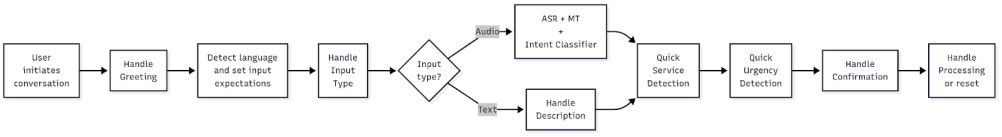
   This model is used by the system to classify the user's intent once the translated text is available. After converting the user's free-form query into a dense numerical meaning vector, it compares its angles to previously calculated vector embeddings of service categories. Even if the user's phrasing is unusual, the model determines the most semantically aligned category by computing cosine similarity. Scalable real-world deployment across a variety of dialects and speech patterns is made possible by this zero-shot classification method, which guarantees robustness and flexibility without the need for extra supervised training.



4. **Chatbot:**
   Natively developed

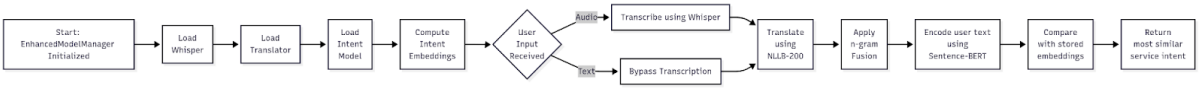
   This part of the system is built with a clear, rule-based structure to handle a variety of natural inputs and is intended for multilingual and low-literacy environments. The following steps are taken in the interaction:

   a. **Input Type Detection**: Before determining the proper processing path, the chatbot determines whether the user is interacting by text or voice.
   b. **Language Detection**: It determines whether the input contains Hindi, Tamil, or English using straightforward Unicode range-based rules.
   c. **Intent and Urgency Analysis**: Whisper is used to transcribe voice input, and NLLB-200 is used to translate it. Sentence-BERT then uses semantic similarity to classify the service intent, and a keyword-based rule system determines urgency.
   d. **Confirmation and Feedback**: After summarizing the identified request and its urgency, the chatbot requests confirmation from the user and either allows corrections or moves forward with simulated processing.



5. **Model Manager Functions**:
   Natively developed

   a. **_load_whisper()**: Loads the Whisper model for converting audio input into text. It prepares the model for inference on 16 kHz mono audio and supports mixed-language transcription.
   b. **_load_translator()**: Initializes the NLLB-200 translation pipeline to convert regional language segments (Hindi/Tamil) in the text into fluent English.
   c. **_load_intent_model()**: Loads the Sentence-BERT model used to generate sentence embeddings for user queries and service categories.
   d. **_compute_intent_embeddings()**: Encodes predefined service categories into vector form using Sentence-BERT and stores them for quick semantic matching during runtime.
   e. **apply_ngram_fusion()**: Applies simple rule-based keyword corrections to fix minor errors in transcriptions (e.g., mapping “plambing” to “plumbing”).



Pipeline Explanation:

The cascaded modular architecture of EchoLang is intended to manage multilingual user inputs in Tier 2/3 Indian real-world settings. There are four main steps in the pipeline's processing of user speech:

1. **Audio Input and Preprocessing**: Wav files are required for audio inputs. To satisfy Whisper's specifications, all input waveforms are automatically resampled from 32 kHz to 16 kHz using scipy.signal.resample. After being transformed into a NumPy array, the waveform is sent straight to the transcription engine.

2. **Automatic Speech Recognition (ASR)**: The audio is transcribed using OpenAI's Whisper (medium) model. It can handle background noise and dialect variation with high robustness and supports code-switched inputs in Tamil, Hindi, and English. Additionally, Whisper recognizes the most common spoken language without the use of explicit language tags.

3. **Postprocessing and N-gram Fusion**: After transcription is finished, frequent low-level substitution or spelling errors can be fixed with optional n-gram-based keyword correction. Using a pseudo dictionary, this helps refine terms like "elektrician" to "electrician." The final pipeline does not use neural translation.

4. **Intent Classification (SBERT)**: Sentence-BERT (all-MiniLM-L6-v2) is used to embed the clean English text, and cosine similarity is calculated against a predetermined embedding bank of service categories. The intended service type is chosen from the category with the highest degree of similarity. Keyword matching is used in parallel to classify urgency (e.g., "immediately," "urgent").

This structure allows EchoLang to be robust with handling informal phrasing and multilingualism.

Model Performance:

Tested Using Google FLEURS (English, Hindi and Tamil Datasets)

English:

```
English Results:
Samples: 50
WER: 0.2488
CER: 0.0641

English Sample Predictions:
                                                Reference                                                 Prediction
however due to the slow communication channels styles in the...  However, due to the slow communication channels, styles in ...
all nouns alongside the word sie for you always begin with a...  All nouns alongside the world say for you always begin with...
to the north and within easy reach is the romantic and fasci...  To the north and within easy reach is the romantic and fasc...
```

Hindi:

```
Hindi Results:
Samples: 50
WER: 0.4705
CER: 0.2237
```

```
Hindi Sample Predictions:Reference                                                    Prediction
कुछ अणुओं में अस्थिर केंद्रक होता है जिसका मतलब यह है कि उनम...  कुछ अड़ोंगों में आर्सत्र केंद्रक होता है जिसका मतलब यहां कि...
ग्रीनलैंड को बहुत कम जगह बसाया गया था नॉर्स सगास में वे कहते...  ग्रीनलेंड को बहुत कम जगव बसाया गया था नौर्स सगास में वे कहत...
ऐसी कोई वैश्विक परिभाषा नहीं है जिसके लिए निर्मित सामान एंटी...  ऐसी कोई वस्पिक परिवाशा नहीं हैं, जिसके लिए निर्मित समान अन्...
```

Tamil:

```
Tamil Results:
Samples: 50
WER: 0.6478
CER: 0.2399
```

```
Tamil Sample Predictions:
                                                    Reference                                                    Prediction
இது வேதியியல் ph என அழைக்கப்படுகிறது நீங்கள் சிவப்பு முட்டைக...  இது வேதியில் பிகச்சென அழைக்கப்படுகிறது நீங்கள் சிவுப்பு முட...
லக்கா சிங் பஜனையை வழங்கினார் பாடகர் ராஜு கண்டெல்வலும் அவருடன...  లక్కాజింగ్ బజన్ వాలంగినార పాడగర రాజు కండేల వాలు అవరిదంన్న వాంగి...
நீங்கள் உங்கள் சொந்த கருத்தைத் தவிர அரசாங்கங்களின் ஆலோசனையை ...  நீங்கள் உங்கள் சொந்த கருத்தைத் தவிர, அரசாங்கங்களின் ஆலோசனைய...
```

For each of the supported languages - English, Hindi, and Tamil - 50 sample audio clips were used to assess EchoLang's ASR pipeline. Due to acoustic overlap and variations in pronunciation, the model's performance for lower-resource languages decreased slightly, but overall it demonstrated strong transcription accuracy across languages. English had the lowest Word Error Rate (WER) and Character Error Rate (CER), at 0.2488 and 0.0641, respectively. This suggests that high-resource language inputs have high transcription accuracy. With WER = 0.4705 and CER = 0.2237, Hindi samples performed moderately, illustrating the difficulties caused by intricate scripts and frequent code-mixing. Despite having the highest error rates (WER = 0.6478 and CER = 0.2399), Tamil, a language with a rich phonetic and morphological structure, maintained significant semantic structure in the majority of predictions.

Tools, Libraries Used:

OpenAI Whisper (whisper-medium), Meta NLLB-200, Sentence-BERT (all-MiniLM-L6-v2), Torch, PyTorch, SciPy (scipy.signal), NumPy, torchaudio, regex, sentence-transformers

---

References:

- Adapting Whisper for Code-Switching through Encoding Refining and Language-Aware Decoding
- Enhancing Whisper's Accuracy and Speed for Indian Languages through Prompt-Tuning and Tokenization
- Turning Whisper into Real-Time Transcription System
- Speech Recognition Based Prescription Generator
- NICT's Cascaded and End-To-End Speech Translation Systems using Whisper and IndicTrans2 for the Indic Task