

Code-Switching-Aware Speech Recognition and Selective Translation for Telugu-English in Legal and Medical Domains

B. Pravalika¹, Emani Saketh², E. Vineela³, Lakshmi Prasanna Palagani⁴

¹Assistant Professor & ²³⁴Student, Hyderabad Institute of Technology and Management
Department of Computer Science and Engineering, [HITAM], [Telangana], India

ABSTRACT - People rarely stay in one language when they speak. In a courtroom in Telangana, a sentence may begin in Telugu, drop in an English legal term and a section number, and end in English; a patient in a clinic may say that they took a fever tablet in the middle of an otherwise Telugu sentence. Most speech recognition and live-translation tools assume one language per utterance. They either garble the switched words or convert everything into English, which throws away exactly the terms that matter most. This paper describes the design and first experimental baselines of a code-switching-aware speech recognition and selective translation system for Telugu-English speech in the legal and medical domains. The proposed pipeline transcribes speech in a mixed-script form (Telugu words in Telugu script, English words in English, numbers as digits), tags each segment by language, protects domain terms, and sends only the segments that need translation to a translation model. As a first step we fine-tuned two open ASR models. Whisper Base, trained on about 26.9 hours of Telugu and English speech, reached a 6.07% word error rate (WER) on a 713-utterance English test set but 45.25% on a 903-utterance Telugu validation set. A wav2vec2-base model trained with CTC on about 14 hours of speech reached a 39.26% validation WER after 19 epochs. Neither model was stable enough for court or clinical use, and neither has yet been tested on code-switched speech, which is the gap the proposed system is designed to close.

Key Words: Automatic Speech Recognition (ASR), Code Switching, Telugu, Whisper, wav2vec 2.0, Connectionist Temporal Classification (CTC), Legal Speech, Medical Speech, Selective Translation, Word Error Rate.

1. INTRODUCTION

Speech is becoming a primary input for documentation. Court proceedings need accurate records, doctors need consultations written down, and online meetings increasingly come with live captions. In India these settings are almost never monolingual. Telugu speakers

routinely mix in English, and the English they use is not random: it tends to be the legal term ("bail", "section 144", "affidavit"), the medical term ("fever", "BP", "tablet") or a number. Linguists call this code-switching, and it is the normal way many people talk rather than an error to be corrected.

Automatic speech recognition (ASR) has improved sharply with large pretrained models such as wav2vec 2.0 [2] and Whisper [1], but these models are usually trained, evaluated and deployed with one language per utterance in mind. A system that hears "meru ivala section 144 lo unaru you may get arrested" has to decide, word by word, which language it is hearing, write each word in the right script, and keep the legal reference intact. Getting this wrong in a courtroom or a clinic is not a cosmetic problem, because a mistranscribed section number or drug name can change the meaning of a record.

1.1 Problem Statement

Given an audio stream containing Telugu speech with English words and numbers mixed in, the system should produce a transcript in which each word appears in its own language and script, and, where a translation is requested, only the words that need translating are translated. Table -1 shows the behaviour we are aiming for. The transcripts are written as we intend them to appear; the Telugu script is shown in Telugu, and English words and numbers are left untouched.

Table -1: Illustrative Target Behaviour (Legal and Medical)

Spoken input (romanised)	Intended mixed script transcript	What must be preserved
meru ivala section 144 lo unaru you may get arrested	మీరు ఇవాళ section 144 లో ఉన్నా రు you	Section number as digits; English clause kept in English

	may arrested	get (legal)
nenu fever tablet vesukuna	నేను ఇవాళ fever tablet వేసుకున్నా	"fever" and other medical terms kept in English (medical)

Existing tools handle this poorly. General-purpose ASR tends to force one language onto the whole utterance, and online meeting tools that offer live translation convert everything into a single target language, usually English, so the mixed terms that carry the meaning are lost or rewritten.

1.2 Motivation

The project began as part of ongoing ASR research on legal and medical speech at IIIT Hyderabad. Our early experiments with off-the-shelf and fine-tuned models made the gap concrete: models that looked reasonable on clean English fell apart on Telugu, and none of them had ever been asked to handle both in one sentence. Legal and medical speech makes the problem harder, because domain vocabulary is dense, numbers matter, and errors are costly. The same idea also extends beyond these two domains. Multilingual meetings, for example, would benefit from captions that keep each speaker's mix of languages instead of flattening it.

1.3 Objectives of the Project

- Build and evaluate bilingual Telugu–English ASR baselines by fine-tuning open models (Whisper and wav2vec 2.0) on Telugu and English speech.
- Produce transcripts in mixed-script form, with Telugu words in Telugu script, English words in English, and numbers as digits.
- Tag each segment of the transcript by language so that later stages can treat it correctly.
- Protect legal and medical terms from being mistranscribed or mistranslated.
- Translate only the segments that need translation, so that multilingual output (for example, live captions in a meeting) keeps the speaker's own mix of languages.

1.4 Scope of the Project

This paper covers the first phase of the work. The language pair is Telugu and English, and the target domains are legal and medical speech. What has been implemented and measured so far is the ASR baselines: two fine-tuned models, evaluated on monolingual Telugu and English data. The language tagging, domain-term protection, selective translation and live-captioning components are described as the proposed design and have not yet been built or evaluated. The results reported here should therefore be read as baselines that motivate the design, not as results for the complete system.

1.5 Organization of the Paper

Section 2 reviews existing approaches and where they fall short. Section 3 describes the data, the two baseline models and the proposed code-switching stage. Section 4 presents the system architecture. Section 5 reports the experimental results. Section 6 discusses the difficulties we ran into, and Section 7 concludes with future work.

2. LITERATURE SURVEY

Table -2 summarises the model families and tools most relevant to this work and shows where each falls short for Telugu–English code-switched speech in the legal and medical domains.

Table -2: Existing Approaches Relevant to Code-Switched ASR

Platform	What It Does	Key Features	Where It Falls Short
Whisper [1]	Multilingual encoder–decoder ASR trained on large scale weakly supervised data	Robust to accents and noise; one model covers many languages	Accuracy drops on lower resource languages such as Telugu; language is normally set per decoding window, which is awkward for switching inside a sentence

wav2vec 2.0 [2] with CTC [3]	Self-supervised speech encoder fine-tuned with a character-level CTC head	Works with modest, labelled data; small enough to fine tune on one GPU	Needs a language-specific character vocabulary; without a language model the word error rate stays high; no built-in language identification
Indian language ASR benchmarks and models [4]	Benchmarks and fine-tuned models for major Indian languages	Covers Telugu and other Indian languages with shared evaluation sets	Largely monolingual and general domain; not built for legal or medical vocabulary or for switching inside one sentence
Live meeting captioning and translation tools [5]	Produce captions or translated captions during online meetings	Real-time and easy to use	In our use, they translate whole utterances into one target language, so English terms inside Telugu speech are not preserved.

2.1 Existing Techniques

Work on code-switched speech, surveyed in [4], has followed a few broad routes: training one multilingual model on a merged vocabulary, adding language identification to the recogniser, adapting the language model with code-switched text, and generating synthetic code-switched data when real recordings are scarce. These approaches have been studied mostly on other language pairs and in general conversation. For Telugu-English in legal and medical settings, we are not aware of a ready-to-use system that combines mixed-script transcription with selective translation.

Machine translation for Indian languages has also advanced. Models such as IndicTrans2 [9] cover the 22 scheduled Indian languages and are a natural candidate for the translation stage we propose. The open question is not

whether translation is possible but how to apply it only to the segments that need it, without touching protected terms.

3. FRAMEWORK / METHODOLOGY

Our method has two parts. The first, which is implemented, is a pair of fine-tuned ASR baselines. The second, which is proposed, is a code-switching stage that sits on top of the recogniser. Figure -1 shows the complete flow from spoken input to mixed-language output.

3.1 Data

The Whisper experiments used a combined Telugu and English corpus. The target was about 15 hours of each language; the prepared data came to about 11.90 hours of Telugu (7,216 WAV files, 7,215 paired records) and 15.00 hours of English, about 26.90 hours in total. Evaluation used a 1.5-hour Telugu validation set (903 WAV files with transcripts) and a 1.5-hour English test set (713 FLAC files with transcripts). The wav2vec2 experiments used a separate dataset of 5,705 training utterances (about 14 hours) and 634 validation utterances, all at 16 kHz. None of these datasets contains code-switched speech.

3.2 Baseline A: Fine-Tuned Whisper Base

We fine-tuned the Whisper Base model on the combined corpus for 3 epochs (3,915 training steps) on an NVIDIA T4 GPU in Google Colab. The final validation loss was 0.104132, and checkpoints were saved at steps 3000, 3500 and 3915; the last is used for all reported results. For Telugu evaluation, decoding was set to the Telugu language and the transcription task.

3.3 Baseline B: Fine-Tuned wav2vec2-base with CTC

We fine-tuned facebook/wav2vec2-base (94.45 million parameters, of which 90.25 million were trainable) with CTC loss and a custom character-level vocabulary. The convolutional feature encoder was frozen. To fit a 6 GB GPU, we used a physical batch size of 2 with gradient accumulation over 4 steps (effective batch size 8), mixed-precision (FP16) training, and expandable CUDA memory segments. Training was resumed from a saved checkpoint at epoch 8 and continued to epoch 19, with validation WER recorded after each epoch.

3.4 Proposed Code-Switching Stage

On top of the recogniser we propose four steps, none of which is evaluated yet:

- Mixed-script transcription. The recogniser is trained to write Telugu words in Telugu script and English words in English, with numbers as digits, so that "section 144" is written as it is spoken and understood.
- Language tagging. Each word or segment is labelled as Telugu, English or number, so that later stages know what to do with it.
- Domain term protection. A legal and medical term list, extended over time, marks words that must be kept exactly as recognised and never translated or rewritten.
- Selective translation. Only segments tagged as needing translation are sent to a translation model (for example, IndicTrans2 [9]). Everything else is passed through, which is what keeps the speaker's own mix of languages intact in a multilingual meeting.

3.5 Evaluation Metrics

We report word error rate (WER), and character error rate (CER) where available, as the primary measures. WER is the number of substituted, deleted and inserted words divided by the number of words in the reference. As code-switched test data becomes available, we plan to add language-tagging accuracy, the fraction of protected terms preserved correctly, and a mixed error rate that counts words and characters in a way that suits mixed-script text.



Fig -1: Process Flow of the Proposed Code-Switching-Aware ASR and Selective Translation System

4. ARCHITECTURAL DESIGN

The system is organised in five layers: Presentation, Audio, ASR & Language, Translation & Post-processing, and Data. Each layer has one job, so a component can be replaced without disturbing the others; for example, the recogniser can be swapped from Whisper Base to a larger Whisper model, or the translation model can be changed, without touching the interface. Figure -2 shows the layers and how they connect.

4.1 Presentation Layer

This is the part a user sees: a simple web interface, later a plug-in for meeting tools, that shows the live captions and lets the user export the transcript. In a court or clinic use, it would show the mixed-language transcript; in a multilingual meeting, it would show captions in which each participant's own mix of languages is kept.

4.2 Audio Layer

Speech is captured from a microphone, a recording or a meeting stream, resampled to 16 kHz (the rate both baseline models expect), and cut into segments at pauses so that each segment can be recognised separately.

4.3 ASR and Language Layer

This layer holds the fine-tuned Whisper and wav2vec2 recognisers described in Section 3, plus the proposed language tagger. Whisper and wav2vec2 are kept side by side so that they can be compared on the same code switched data once it exists.

4.4 Translation and Post-processing Layer

Here the domain term list is applied, numbers and named entities are handled, and only the segments that need translation are sent to the translation model. The output is the final mixed-language transcript or caption.

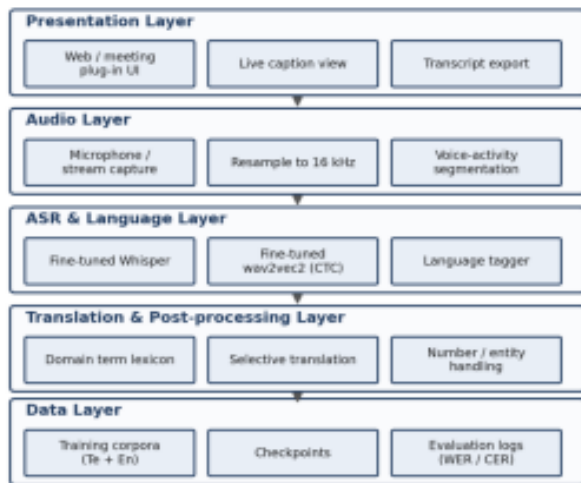
4.5 Data Layer

Training corpora saved model checkpoints and evaluation logs are stored here. In our experiments, checkpoints and evaluation outputs (including the per-utterance English predictions) were stored on Google Drive so that every reported number can be reproduced.

4.6 Data Flow across the Architecture

Following one request end to end: audio is captured and

resampled, split into segments, and recognised into mixed script text; each word is tagged by language; protected legal and medical terms are marked; only the segments that need it are translated; and the interface shows the transcript or live caption. Table -3 lists the technologies used or planned for each layer.



Layer	Component	Technology Used
ASR & Language	Job Description Parser	Python string/regex utilities
ASR & Language	Text Preprocessing	NLTK (tokens, stop-words, lemmas)
Translation	Keyword/Skill Extraction	spaCy (Named Entity Recognition)
Presentation	Semantic Similarity	Scikit-learn (TF-IDF), SBERT
Data	ATS Scoring Engine	Custom Python weighted scoring logic

5. ANALYSIS

Table -4 summarises the two experimental setups, and Table -5 gives the results. Chart -1 compares the final error rates and Chart -2 shows how the wav2vec2 validation WER improved over training.

Table -4: Experimental Setup of the Two Baselines

Item	Baseline A: Whisper Base	Baseline B: wav2vec2-base
Training data	≈ 26.90 h (11.90 h Telugu + 15.00 h English)	5,705 utterances (≈ 14 h)
Evaluation data	Telugu: 903 utt. (1.5 h); English: 713 utt. (1.5 h)	634 validation utterances
Training length	3 epochs, 3,915 steps	Epochs 8 to 19 (resumed)
Hardware	NVIDIA T4 (Google Colab)	Single 6 GB NVIDIA GPU
Objective / head	Sequence-to-sequence decoder	CTC, character-level vocabulary

Table -5: Baseline Results

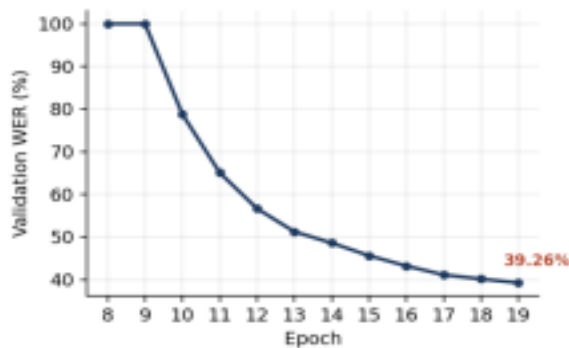
Model / evaluation set	WER	CER	Samples
Whisper Base:English test	6.07%	2.03%	713
Whisper Base: Telugu validation	45.25%	Not reported	903
Wav2vec2-base: validation(epoch 19)	39.26%	Not reported	634

The Whisper model does well on English, at 6.07% WER and 2.03% CER, but its Telugu WER of 45.25% is far higher. That gap is what we would expect if English is much better covered by the model's pretraining, though we have not tested this directly. A Telugu example from the evaluation shows the kind of errors involved. For the reference చింటా అలియాస్ చింద్రశేఖర్ చింద్రశేఖర్ నివాసింల్ పోలీసులు సోదా చేసిన the model produced చింటాల్యాస్

చింద్రశేఖర్ నివాసను పోలీసులు సోదా చేసి, with a spelling variation in "alias", one omitted repeated name, and a changed word near the end. Most of the sentence was right, but the errors fall on exactly the kind of words (names, places) that matter in legal text.

For wav2vec2, validation WER stayed at 100% for the first two recorded epochs, fell to 78.79% at epoch 10, dropped below 50% at epoch 14, and reached 39.26% at epoch 19, an improvement of 60.74 percentage points over the epoch 8 checkpoint. Validation loss fell from 358.76 to 34.21 over the same period, so the improvement was steady and there is no sign of overfitting within the recorded window. Memory saving measures (small physical batch, gradient accumulation, FP16) made training on a 6 GB GPU feasible, with only occasional gradient-overflow warnings early in training.

Chart -1: Final Word Error Rate of the Baselines



5.1 What These Results Do and Do Not Show

- The two baselines were trained and evaluated on different data, so their WERs should not be compared as a ranking of the two models.
- A WER of 45.25% (Whisper, Telugu) and 39.26% (wav2vec2) is still too high for use in a courtroom or clinic, where a wrong name, number or drug matters.
- Every number above is on monolingual data. None of them says how the models behave on code-switched speech, which is the case this work is about.
- The wav2vec2 result comes from a single validation split; a held-out, speaker-independent test set is needed before drawing broader conclusions. For Whisper, a breakdown of substitutions, deletions and insertions can be computed from the saved predictions but is not reported here.

6. CHALLENGES

- No code-switched Telugu–English legal or medical speech corpus was available to us, so both baselines were trained and tested on monolingual data and the code switching behaviour is still untested.
- Telugu accuracy remained the weak point. Spelling variation and word boundaries in Telugu script make WER harsh, and small differences (for example in how a word like "alias" is written) are counted as errors.
- The Whisper training data was smaller than planned for Telugu: about 11.90 hours against a 15-hour target, and less than the 15.00 hours of English, so the two languages were not balanced.
- Hardware limits shaped the wav2vec2 experiments. A 6 GB GPU forced a batch size of 2 with gradient accumulation, and early FP16 gradient overflows had to be tolerated until the loss scale settled.
- Legal and medical speech mixes vocabulary and numbers. Section numbers, drug names and abbreviations have to be transcribed exactly, which ordinary WER-driven training does not specifically reward.
- Deciding what should stay in English and what should be written in Telugu is itself a design question, because speakers differ in which English words they use, and a fixed word list will not cover everything.

7. CONCLUSIONS

This paper set out to make speech technology work for the way people in Telangana actually speak, with Telugu and English mixed together, in settings such as courts and clinics where accuracy matters. We described a code-switching-aware pipeline that writes each language in its own script, tags segments, protects domain terms and translates only what needs translating. As a first step we built two ASR baselines. Whisper Base reached 6.07% WER on English and 45.25% on Telugu, and wav2vec2-base reached 39.26% WER after 19 epochs. Neither is stable enough for real use, and neither has been tested on code-switched speech, but together they give us a measured starting point.

The next steps are clear. We plan to collect or build a code switched Telugu–English legal and medical test set, try larger Whisper models, add a language model to the wav2vec2 decoder (for example, an n-gram model) and measure the change, implement the language tagger and term list, and evaluate the full pipeline with the code-switching measures described in Section 3.5. Longer term,

the same approach could be applied to other domains and to live multilingual captions in online meetings.

and Accessible Machine Translation Models for all 22 Scheduled Indian Languages," Transactions on Machine Learning Research, 2023.

ACKNOWLEDGEMENT

We are grateful to B. Pravalika and the [LRCT] at IIIT Hyderabad for guidance and support throughout this research internship.

REFERENCES

- 1) A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," arXiv:2212.04356, Dec. 2022.
- 2) A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020.
- 3) A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in Proc. 23rd Int. Conf. on Machine Learning (ICML), 2006, pp. 369–376.
- 4) S. Sitaram, K. R. Chandu, S. K. Rallabandi, and A. W. Black, "A Survey of Code-switched Speech and Language Processing," arXiv:1904.00784, 2019.
- 5) K. Bhogale et al., "Vistaar: Diverse Benchmarks and Training Sets for Indian Language ASR," in Proc. Interspeech, 2023.
- 6) T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proc. 2020 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.
- 7) A. Paszke et al., "PyTorch: An Imperative Style, High Performance Deep Learning Library," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019.
- 8) P. Mikićević et al., "Mixed Precision Training," in Proc. Int. Conf. on Learning Representations (ICLR), 2018.
- 9) J. Gala et al., "IndicTrans2: Towards High-Quality