

RAILVAANI: An Optimized Bilingual ASR Pipeline for Entity-Rich Railway Announcements

Anand Kumar Rai
IIT Kharagpur

Swar Desai
IIT Kharagpur

Shshank Mittal
IIT Kharagpur

Animesh Mukherjee
IIT Kharagpur

Abstract

Automatic Speech Recognition (ASR) for railway announcements is challenging due to noisy environments and dense, entity-rich content. We introduce RAILVAANI, a low-latency bilingual (Hindi-English) ASR pipeline for Indian railway public address (PA) announcements, benchmarked on **RailKosh** — the first bilingual annotated corpus of real railway PA audio (1,513 clips: 1,000 English, 513 Hindi; 53 stations; gold transcripts and entity labels). We evaluate ASR baselines and their LoRA fine-tuned variants with database-aware (DB-aware) post-processing (PP), using WER and *task score* (mean of train number accuracy, platform accuracy, and announcement-type macro-F1), as task score is the more passenger-relevant metric for this domain. Our key findings: (i) on English PA, our deployed system (Whisper Small + LoRA + PP) achieves 69.8% task score at a real-time factor (RTF) of 0.75; a larger variant (Qwen3-ASR-0.6B + LoRA + PP) matches Google STT (79.7% vs. 79.3%) at RTF=1.37; (ii) on Hindi PA, our deployed system (collabora/whisper-small-hindi + LoRA + PP) achieves 60.7% task score at RTF=0.83, surpassing whisper-large-v3 (43.5%) with a 6× smaller model; and (iii) a user study ($N=63$, SUS-style median=90.0) indicates strong perceived usability. Both deployed systems run faster than real-time on CPU, enabling deployment without GPU infrastructure. RAILVAANI, **RailKosh**, and the RailKosh annotation toolkit are fully open-source.

1 Introduction

Railway public address (PA) announcements serve as the primary real-time communication channel at Indian railway stations, conveying critical information — train arrivals, departures, platform assignments, delays, and cancellations — to millions of passengers daily. Yet passengers frequently miss or misinterpret these announcements. PA systems introduce reverb, frequency distortion, and

background noise that significantly degrade speech intelligibility, posing particular challenges for elderly passengers, hearing-impaired travellers, non-native speakers, and first-time users (Gregori et al., 2023; Kumar et al., 2020; Sapkota and Shrestha, 2020).

Indian Railways serves over 23 million passengers daily across approximately 7,000 stations (Kumar et al., 2020). Public announcements are delivered through the Integrated Passenger Information System (IPIS) (Research Designs and Standards Organisation (RDSO), 2021), automatically generating announcements by concatenating pre-recorded audio segments at runtime and playing them through the station PA speakers. Automating the understanding of these announcements can significantly benefit hearing-impaired travellers, the elderly passengers, and non-native speakers who struggle to parse them in noisy environments (Gregori et al., 2023; Strik et al., 1997).

However, current standard benchmarks such as Whisper and QwenASR (Radford et al., 2023; Shi et al., 2026) struggle to transcribe railway PA audio reliably. The stitched nature of IPIS audio introduces prosodic discontinuities at splice points between concatenated segments, while low-fidelity PA speakers in crowded environments further degrade intelligibility. Announcements are also entity-rich; a single 20-second clip contains a train number, train name, origin, destination, via stations, platform number, and delay status — yet embedded in formulaic filler phrases (“*May I have your attention please...*”) irrelevant to passenger action, causing standard WER to overstate system failure on practically important content. While domain adaptation (Hsu et al., 2017; Sim et al., 2024), LoRA fine-tuning (Hu et al., 2022), and post-processing against domain knowledge bases (Tang et al., 2024) have shown promise in specialised ASR settings, they remain unexplored for Indian railway PA audio, where entity vocabularies

span thousands of station names, train numbers, and regional language variants.

We introduce **RAILVAANI**¹, a bilingual (Hindi-English) end-to-end PA understanding pipeline comprising three components: (i) domain-adapted ASR modules²³ for Hindi and English PA announcements running on CPU via faster-whisper for low-latency real-time inference; (ii) a DB-aware entity extraction module supporting train number, platform, and route extraction in both Devanagari and Roman scripts with edit-distance correction against a curated train number database⁴; and (iii) a MuRIL-based (Khanuja et al., 2021) announcement type classifier⁵ covering arrival, departure, standing, delay, and other categories. To support systematic evaluation, we introduce **RailKosh**⁶, the first bilingual annotated corpus of real Indian railway PA audio (1,513 clips, 53 stations, gold transcripts and entity labels), and the **RailKosh Annotation Toolkit**⁷, a zero-install tool with silver transcript pre-filling, multilingual input support, and automatic inter-annotator agreement computation.

We evaluate various ASR systems on **RailKosh** using WER and *task score* (mean of train number accuracy, platform accuracy, and announcement-type macro-F1) on English ($n=1,000$) and Hindi ($n=513$) test sets. **RAILVAANI** surpasses whisper-large-v3 on both languages, achieving 79.7% task score on English ($2.5\times$ smaller model) and 60.7% on Hindi ($6\times$ smaller model), matching Google STT on English task score while running entirely on CPU. A user study ($N=63$, SUS-style score median = 90.0) indicates strong perceived usability. **RailKosh** and the train/station database are released under CC-BY-4.0, while **RAILVAANI**, its ASR models, the announcement classifier, and the Annotation Toolkit are released under MIT licence.

¹<https://huggingface.co/spaces/raianand/RailVani>

²<https://huggingface.co/raianand/collabora-whisper-railvaani-ft>

³<https://huggingface.co/raianand/railvaani-whisper-small-ft>

⁴<https://huggingface.co/datasets/raianand/RailVaani-StationTrainDict>

⁵<https://huggingface.co/raianand/railvaani-muril-classifier>

⁶<https://huggingface.co/datasets/raianand/RailKosh>

⁷<https://github.com/raianand1991/railkosh-toolkit>

Statistic	English	Hindi
Total clips	1,000	513
Total duration	4.4h	3.1h
Unique trains	385	304
Unique stations	414	238
Avg clip duration	15.8s	21.8s
Median SNR(dB)	-0.6	0.4
<i>Announcement type</i>		
Arrival	594 (59.4%)	206 (40.2%)
Departure	205 (20.5%)	96 (18.7%)
Standing	98 (9.8%)	116 (22.6%)
Delay	92 (9.2%)	39 (7.6%)
Other	11 (1.1%)	56 (10.9%)

Table 1: **RailKosh** corpus statistics. Hindi clips are longer on average (21.8s vs 15.8s) due to more complex announcement structures. Other includes ‘Cancelled’, ‘Rescheduled’, and ‘Platform Change’. Unique stations count all ‘From’, ‘To’, and ‘Via’ stations across the corpus.

2 Dataset

2.1 RailKosh data collection

RailKosh comprises 1,513 real PA audio clips (4.4 hours English, 3.1 hours Hindi) recorded across 53 Indian railway stations (list is provided in Annexure). Audio was collected from two sources: direct station recordings by the authors spanning major junction stations like New Delhi, Chandigarh, Howrah, and publicly available railway videos from YouTube channels⁸. This combination captures natural variation in PA audio quality, PA system characteristics, and announcement styles across the Indian Railways network.

The detailed statistics are presented in Table 1. **RailKosh** clips exhibit near-zero SNR across both languages (English: -0.6 dB, Hindi: 0.4 dB median), reflecting the challenging acoustic conditions of real PA environments where background noise is comparable in power to the speech signal. For ASR evaluation, **RailKosh** is used strictly as a benchmark set: no **RailKosh** audio-transcript pair is used to fine-tune the ASR models.

2.2 Annotation process and toolkit

All clips were manually transcribed and annotated using the **RailKosh Annotation Toolkit**, released as an open-source contribution. Google Speech-to-Text (STT)⁹ outputs served as silver transcripts, pre-filling the interface to accelerate manual correction. Annotators corrected ASR errors and verified five

⁸<https://www.youtube.com/@RailGuruji>; <https://www.youtube.com/@GopiRailworld>.

⁹<https://cloud.google.com/speech-to-text>

structured entity fields per clip: train number, train name, announcement type, platform number, and route (origin, via, destination). Structured fields achieved 100% agreement, as PA announcements follow a fixed template structure that leaves no ambiguity in entity values. Transcript-level agreement was strong (Cohen’s $\kappa = 0.73$), with disagreements resolved through adjudication before finalising the gold transcripts. The toolkit is detailed in Appendix A.

2.3 Synthetic data generation

To address the scarcity of labelled railway PA audio for fine-tuning, we construct a template-driven synthetic data pipeline. Announcement templates covering five types (‘Arrival’, ‘Departure’, ‘Standing’, ‘Delay’, ‘Other’) include placeholders for train number, train name, origin, destination, via stations, platform number, and delay duration, populated using Indian Railways metadata from IndiRailInfo¹⁰ and a Kaggle railway dataset.¹¹ Numerical fields (train and platform numbers) are assembled from concatenated human-recorded digit clips to improve naturalness, mirroring the stitched structure of real IPIS announcements (Research Designs and Standards Organisation (RDSO), 2021).

Audio is synthesised using two TTS engines: Google TTS (gTTS)¹² and Microsoft Azure Neural TTS,¹³ producing 5,088 clips from each engine for a total of 10,176 clips per language (≈ 41 hours English, ≈ 52 hours Hindi).

3 System architecture

RAILVAANI processes raw PA audio through a LORA fine-tuned ASR module, followed by DB-aware post-processing (PP) of the generated transcript via an entity extraction module and an announcement type classifier (Figure 1). The system operates bilingually, unifying Hindi (Devanagari) and English (Roman) under a common output schema.

3.1 ASR module

The ASR module uses faster-whisper for low-latency CPU inference. For English, the deployed system uses Whisper-Small fine-tuned

with LORA (69.8% Task Score); Qwen3-ASR-0.6B with LORA achieves the best English result (79.7% Task Score) but requires GPU and is available as an optional model selection in the demo. We also evaluate whisper-small and other open-source baselines. For Hindi, the deployed system uses a LORA fine-tuned variant of collabora/whisper-small-hindi. The synthetic dataset used for LoRA fine-tune ASR models for both languages is described in Section 2.3.

All fine-tuning uses LORA (Hu et al., 2022) ($r=32$, $\alpha=64$) applied to all attention and feed-forward projection matrices. For English, we fine-tune whisper-small and qwen3-asr-0.6B; for Hindi, we fine-tune collabora/whisper-small-hindi. All models are trained on their respective 10,176 synthetic clips for 10 epochs with a learning rate of 10^{-4} and batch size of 8.

3.2 Entity extraction

Entities are extracted from the ASR transcript using language-specific rule-based methods. Train numbers are recovered via a multi-stage cascade (digit detection, word-to-digit mapping, edit-distance correction against the curated train database¹⁴). Platform numbers use script-specific regex with digit-word fallback, validated against known platforms per station. The train name and route are retrieved from the database once the train number is validated.

3.3 Announcement type classifier

Keyword rules achieve only 15.9% macro F1 on noisy ASR output, so we fine-tune google/muril-base-cased (Khanuja et al., 2021)¹⁵ for five-way announcement classification: ‘Arrival’, ‘Departure’, ‘Standing’, ‘Delay’, and ‘Other’. Separate English and Hindi classifiers are evaluated using 5-fold cross-validation on RailKosh; all reported AnnType F1 and task score values use out-of-fold predictions.

4 Experiments

4.1 Baseline systems

For English PA evaluation, we benchmark open-source baselines including whisper-large-v3, whisper-small (Radford et al., 2023),

¹⁰<https://indiarailinfo.com/>

¹¹<https://www.kaggle.com/datasets/sripaadsrinivasan/indian-railways-dataset>

¹²<https://pypi.org/project/gTTS/>

¹³<https://azure.microsoft.com/en-us/products/ai-services/ai-speech/>

¹⁴<https://huggingface.co/datasets/raianand/RailVaani-StationTrainDict>

¹⁵<https://huggingface.co/google/muril-base-cased>

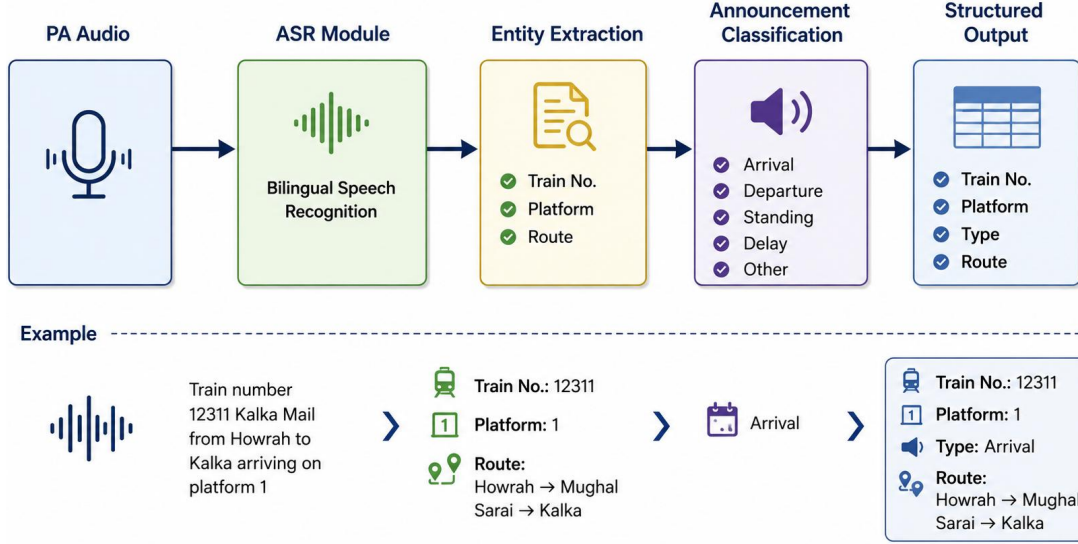


Figure 1: RAILVAANI system architecture. PA audio is processed through three sequential stages: ASR, entity extraction, and announcement classification — producing structured passenger-actionable output.

qwen3-asr-1.7B and qwen3-asr-0.6B (Shi et al., 2026), and conformer-large (Gulati et al., 2020), covering Seq2Seq, LLM-based, and CTC architectures across a parameter range of 120M–1.7B.

For Hindi PA evaluation, we benchmark open-source baseline systems: whisper-large-v3, qwen3-asr-1.7B, and whisper-small, alongside two Hindi fine-tuned Whisper variants: collabora/whisper-small-hindi¹⁶ (whisper-small fine-tuned on 3,000 hours of Hindi speech) and ART-PARK whisper-large-v3-vaani-hindi¹⁷ (whisper-large-v3 fine-tuned on 718 hours of Hindi speech from project Vaani (Pulikodan et al., 2026), covering 773 Indian districts).

Google Speech-to-Text Chirp2¹⁸ serves as the commercial API reference across both English and Hindi evaluations.

4.2 Task score: A domain-appropriate metric

Standard WER is inflated on PA audio by filler phrases and Roman-script station names in predictions mismatching Devanagari gold references. We define

Task score = $\frac{1}{3}(\text{TN_Acc} + \text{Plat_Acc} + \text{Type_F1})$ where TN_Acc is exact-match train number accuracy, Plat_Acc is exact-match platform accuracy,

¹⁶<https://huggingface.co/collabora/whisper-small-hindi>

¹⁷<https://huggingface.co/ARTPARK-IISc/whisper-large-v3-vaani-hindi>

¹⁸<https://cloud.google.com/speech-to-text>

and Type_F1 is macro-averaged F1 over five announcement types. Task score directly measures whether a passenger can board the correct train on the correct platform.

4.3 Results

English PA (Table 2). The qwen3-asr-0.6B + LORA + PP setup achieves 79.7% task score, surpassing Google STT (79.3%) without cloud API calls, enabling privacy-preserving, offline-capable deployment at stations with intermittent connectivity. LoRA fine-tuning yields the largest WER reduction across all systems (whisper-small: −12.9pp, qwen3-asr-0.6B: −2.7pp), while DB-aware PP adds a consistent +5.4-point task score gain across all architectures 5. For latency-constrained deployment, whisper-small + LORA + PP offers the best speed-accuracy tradeoff (RTF¹⁹=0.75, 69.8% task score).

Hindi PA (Table 3). LoRA fine-tuned collabora/whisper-small-hindi with DB-aware PP achieves 60.7% task score at RTF=0.83—the best open-source result—outperforming whisper-large-v3 (43.5%) with 6× fewer parameters. qwen3-asr-1.7B collapses to 11.2% task score and is also the slowest system tested (RTF=5.28), suggesting LLM-based ASR is poorly suited to Hindi PA acoustics in both accuracy and latency. Google STT (Task Score: 66.7%) remains the ceiling, a 6-point gap to the best open-source

¹⁹<https://openvoice-tech.net/index.php/Real-time-factor>

System	WER↓	TN↑	Plat↑	AnnF1↑	Task↑	Lat.(s)↓
— Commercial API —						
Google STT Chirp2	45.0	78.5	83.0	76.3	79.3	NA
— Open-source baselines —						
whisper-large-v3	35.0	63.3	70.6	75.3	69.7	56.2
qwen3-asr-1.7B	28.9	79.0	83.6	78.7	80.4	45.4
whisper-small	52.1	40.3	64.4	65.1	56.6	11.5
conformer-large	58.1	48.6	56.0	64.7	56.4	3.4
qwen3-asr-0.6B	35.9	69.9	84.6	75.5	76.7	21.3
— LoRA fine-tuned + DB-aware post-processing(PP) —						
whisper-small	39.2	66.4	74.0	69.1	69.8	11.8
conformer-large	55.7	49.6	49.8	62.9	54.1	3.7
qwen3-asr-0.6B	33.2	76.3	82.4	80.5	79.7	21.6

Table 2: English PA results on **RailKosh** ($n=1,000$). Whisper variants have used faster-whisper implementation. **Lat(s)** is average inference time on CPU.

System	WER↓	TN↑	Plat↑	AnnF1↑	Task↑	Lat.(s)↓
— Commercial API —						
Google STT Chirp2	15.3	80.9	60.9	58.3	66.7	NA
— Open-source baselines —						
whisper-large-v3	73.1	51.5	8.0	71.0	43.5	76.5
qwen3-asr-1.7B	97.1	2.8	3.0	27.9	11.2	115.2
whisper-small	94.6	2.5	0	34	12.2	17.9
collabora	61.2	57.4	10.9	77.6	48.6	17.9
ARTPARK LV3 Vaani	67.5	51.9	11.4	64.8	42.7	76.5
— LoRA fine-tuned + DB-aware post-processing(PP) —						
whisper-small	93.8	26.5	11.9	47.3	28.5	18.2
collabora	65.9	65	46.2	70.9	60.7	18.2
— Ablation of LoRA variants on collabora —						
collabora + VanillaAug	81.9	60.0	56.7	63.9	60.2	18.2
collabora + INFONCE	74.2	61.4	43.8	55.5	53.6	18.2

Table 3: Hindi PA results on **RailKosh** ($n=513$).

system.

4.4 Hindi fine-tuning ablation

TTS-only LoRA fine-tuning of collabora/whisper-small-hindi yields 60.7% task score, marginally below the Google STT (Task Score: 66.7%). We test two interventions to close this gap: PA-noise augmented LoRA, adding spectral shaping and additive noise at SNR=8 dB from real **RailKosh** clips (Task Score: 60.2%); and a contrastive learning approach using INFONCE (Oord et al., 2018), aligning clean and PA-noisy encoder representations of the same utterance (Task Score: 53.6%). Both task scores fell below vanilla LoRA fine-tuning.

Table 3 shows these interventions affect different metrics without closing the gap: PA-noise augmentation improves platform accuracy (56.7% vs. 46.2%) but at higher WER (81.9% vs. 65.9%) and lower train-number accuracy (60.0% vs. 65.0%), netting a near-identical task score (60.2% vs. 60.7%); INFONCE alignment underperforms on all three components (TN: 61.4%, Plat: 43.8%, AnnF1: 55.5%; task score 53.6%). Neither closes

Ann. Type	English	Hindi
Arrival	86.5	77.4
Departure	75.8	82.1
Standing	59.3	68.5
Delay	79	73.5
Other	95.2	51.1

Table 4: Per-type announcement type F1 (%) for English (qwen3-asr-0.6B + LoRA + PP) and Hindi (collabora/whisper-small-hindi + LoRA + PP) systems.

System	Δ TN	Δ Plat	Δ AnnF1	Δ Task
whisper-small + LoRA	+4.6	+3.6	+10.4	+6.2
conformer-large + LoRA	+1.0	+1.1	+8.4	+3.7
qwen3-asr-0.6B + LoRA	+4.7	+6.6	+7.3	+6.2
Avg gain	+3.4	+3.8	+8.7	+5.4

Table 5: Impact of DB-aware post-processing on LoRA fine-tuned systems (English, $n=1,000$).

the gap to Google STT (66.7%), suggesting the synthetic corpus’s limitation lies less in acoustic diversity than in the absence of real disfluencies, splice artefacts, and entity distributions that only annotated real PA audio can supply.

4.5 Error analysis

Hindi ASR remains substantially more challenging than English, with the best Hindi system achieving a task score of 60.7 compared to 79.7 for English. Analysis of the Hindi ablation models shows that substitution errors dominate (50.5–55.8% of edit operations), while insertion and deletion rates remain comparatively low (12–14%), indicating that the models generally preserve utterance boundaries but confuse acoustically similar words. Manual inspection further reveals that these substitutions are concentrated on railway-specific entities, including train names, station names, route mentions, and numerical expressions, whereas the announcement template is recognised consistently. This observation is consistent with the relatively robust announcement type F1 scores (Table 4) and the gains from DB-aware post-processing (Table 5), which primarily improve entity-centric metrics and task score despite limited reductions in WER.

5 RAILVAANI demo and user evaluation

5.1 System interface

The RAILVAANI interface (Figure 2) is accessible at <https://huggingface.co/spaces/raianand/RailVani> (demo video²⁰), accept-

²⁰<https://youtu.be/Ix99jUYswyk>

Live Transcription

Upload a railway announcement audio clip for instant transcription and entity extraction.

ASR Model

☒ Whisper Small
 ☐ Qwen3-ASR-0.6B
 Requires GPU

Station (optional):

e.g. New Delhi

Translate to:

Bengali

Upload Audio

Record

Try a sample:

— pick a clip —

0:00 / 0:18

Process Announcement

Transcript

Detected: English

7.7s

MAY I HAVE YOUR ATTENTION PLEASE TRAIN NUMBER ONE THREE ZERO ZERO SIX PUNJABI FROM AMRITSAR VIA MURADHABAT BAREDIH TO HOWRAH VIA PLATFORM VARANASI IS ARRIVING ON PLATFORM NUMBER ONE

Translation

আমি আপনার মনোযোগ দিতে পারি। অনুগ্রহ করে ট্রেন নম্বর ওয়ান থ্রি শূন্য শূন্য ছয় পানজাবি অমৃৎসর থেকে মুর্আদহাবত বারেদিহি থেকে হাওড়া হয়ে পল্ল্যাটফর্ম বারানসী প্ল্যাটফর্ম এক নম্বর পৌঁছাবে।

TRAIN NUMBER

13006

PLATFORM

1

TRAIN NAME

অমৃতসর হাওড়া মনে

TYPE

আগমন

STATUS

আগত

FROM → VIA → TO

অমৃতসর জং → নুরগিনা জং, আম্ভালা কান্টো জং, সাহরানপুর → হাওড়া জং

Figure 2: RAILVAANI demo interface showing English output of PA announcement with Bengali translation.

ing audio upload or live recording, with ten pre-loaded sample clips (five Hindi, five English) for immediate use. For English, users select between two LORA fine-tuned ASR models — whisper-small- (default, CPU) or qwen3-asr-0.6B (GPU); Hindi audio is always processed with LORA fine-tuned collabora/whisper-small-hindi. Users may also optionally specify a station to scope entity extraction to local trains. The system returns a full transcript and a structured entity card (train number, name, type, status, platform, route), with on-demand translation into eight additional Indian languages (these include Bengali, Telugu, Marathi, Tamil, Gujarati, Kannada, Malayalam, and Punjabi) via Google Translate for broader accessibility. The whole system is implemented as a Flask app on HuggingFace Spaces²¹.

5.2 User study

We conducted a between-subjects study to evaluate whether ASR transcripts help passengers extract information from PA announcements. Of 169 participants, $N=99$ answered at least one clip

(our analysis set): *with_transcript* ($N=48$) or *without_transcript* ($N=51$), of whom 63 completed all 10 clips (5 Hindi, 5 English). For each clip, participants answered three questions—train number, platform, and announcement type—from which we compute a per-clip task score. Respondents spanned native Hindi, Bengali, Gujarati, Telugu, and English speakers, all aged 18 or older. *Task score* shows a consistent positive trend favoring the *with_transcript* condition for both Hindi (+5.5 points) and English (+4.0 points), though neither reaches statistical significance at this sample size (Table 7). Language dominates over transcript condition: mean *task score* is far higher for Hindi clips than English clips (84.5% vs. 63.8%, Mann–Whitney U test, $p < 0.001$). Among the three scored fields, train number has the lowest exact-match accuracy overall (69.9%), while platform accuracy (75.6%) and announcement type accuracy (79.2%) are recovered more reliably. A System Usability Scale (SUS)-style questionnaire (Brooke et al., 1996), completed by 63 of the 99 analysis-set participants, indicates acceptable usability (median score = 90.0). User study responses are available in our dataset²², with detailed analysis in Appendix B. No PII was collected from participants in this study.

6 Conclusion

We presented RAILVAANI, a bilingual ASR pipeline for Indian railway PA announcements, with **RailKosh** (1,513 clips, 53 stations) and its **Annotation Toolkit**. On English PA, our deployed system (whisper-small + LORA + PP) runs at RTF=0.75; a larger variant (qwen3-asr-0.6B + LORA + PP) matches Google STT (79.7% vs. 79.3% task score) at RTF=1.37. On Hindi PA, collabora-whisper-hindi + LORA + PP reaches 60.7% task score at RTF=0.83, within 6 points of Google STT (66.7%). PA-noise and contrastive Hindi ablations do not improve over vanilla LORA + PP, indicating real annotated data—not synthetic augmentation—remains the critical bottleneck. A user study ($N=63$; SUS-style median=90.0) indicates strong perceived usability and reveals a statistically significant Hindi–English comprehension gap among study participants (Task Score: 84.5% vs. 63.8%, Mann–Whitney U test, $p < 0.001$); transcript-condition gains were directionally positive but not statistically significant at this

²¹<https://huggingface.co/spaces>

²²<https://huggingface.co/datasets/raianand/railvaani-study-responses>

sample size. Future work: more Indian languages, real-data based fine-tuning, faster Hindi ASR, and accessibility features for hearing-impaired passengers.

Limitations. **RailKosh** covers 53 stations; generalisation to all 7,000+ Indian railway stations remains to be validated. In particular, DB-aware entity correction (train number and platform matching via edit-distance against *RailVaani-StationTrainDict*) is only characterised for stations and trains present in this database; for out-of-database entities, the pipeline falls back to uncorrected ASR output. We plan to scale *RailVaani-StationTrainDict* to cover all Indian Railways stations and trains and evaluate correction accuracy at that scale in future work. The 6-point Hindi gap to Google STT likely reflects the scarcity of annotated Hindi PA data rather than a fundamental architectural limitation, consistent with our fine-tuning ablation (§4.4) showing that real acoustic characteristics, rather than synthetic scale, close the adaptation gap. Our SUS respondents ($N=63$) are a subset of the full user study ($N=99$) and may not fully represent its demographic spread; the 55+ age group is smaller than other bands across the study overall ($N=60$ clip responses), warranting caution in generalising age-related findings.

Ethics and broader impact

Ethics statement

RAILVAANI is motivated by improving railway accessibility for elderly, hearing-impaired, and non-native language passengers. Audio clips in **RailKosh** were recorded in public railway stations where PA announcements are public broadcasts with no reasonable expectation of privacy; no personally identifiable information is present. Annotation was conducted voluntarily by research team members with full awareness of the study purpose. User study participants provided informed consent and responses were anonymised; only demographic group data (age range, native language, travel frequency) was collected. **RailKosh** is released under CC-BY-4.0 and the RAILVAANI system and *Annotation Toolkit* under MIT licence, supporting open community extension to additional Indian languages and domains.

Broader impact

RAILVAANI targets a concrete accessibility gap: PA announcements are often unintelligible to

hearing-impaired travellers, elderly passengers, and non-native speakers, and no low-cost, open-source system currently bridges this at Indian Railways' scale (23M+ daily passengers). Positive impact includes real-time captioning for hearing-impaired users and structured alerts (platform, delay) deliverable via mobile or station displays without new hardware, since RAILVAANI runs on CPU. We release RAILVAANI and **RailKosh** openly to support further work in this direction, and note that responsible deployment considerations specific to our results (entity-error concentration, the Hindi–English performance gap) are discussed in the limitations section.

References

- John Brooke et al. 1996. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Alessandro Gregori et al. 2023. Automatic speech recognition (asr) for communications with seniors and non-hearing users at public spaces: Speech-to-text technology for live subtitling and accessibility. In *Age-Specific Issues. Language, Spaces, Technologies*, pages 165–189. Peter Lang.
- Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.
- Wei-Ning Hsu, Yu Zhang, and James Glass. 2017. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 16–23. IEEE.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. **Muril: Multilingual representations for indian languages**. *Preprint*, arXiv:2103.10730.
- Rakesh Kumar, Vishal Goyal, and Lalit Goyal. 2020. Railway stations announcement system for deaf. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON): System Demonstrations*, pages 40–42.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Sujith Pulikodan, Abhayjeet Singh, Agneeth Basu, Nihar Desai, Pavan Kumar J, Pranav D Bhat, Raghu Dharmaraju, Ritika Gupta, Sathvik Udupa, Saurabh Kumar, Sumit Sharma, Vaibhav Vishwakarma, Visruth Sanka, Dinesh Tewari, Harsh Dhand, Amrita Kamat, Sukhwinder Singh, Shikhar Vashishth, Partha Talukdar, and 2 others. 2026. *Vaani: Capturing the language landscape for an inclusive digital india*. *Preprint*, arXiv:2603.28714.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Research Designs and Standards Organisation (RDSO). 2021. Integrated passenger information system (IPIS). [https://rdso.indianrailways.gov.in/uploads/files/Pamphlet%20on%20Integrated%20Passenger%20Information%20System%20\(1\).pdf](https://rdso.indianrailways.gov.in/uploads/files/Pamphlet%20on%20Integrated%20Passenger%20Information%20System%20(1).pdf). Ministry of Railways, Government of India. Accessed: July 2026.

Abhijeet Sapkota and Raju Shrestha. 2020. Accessible driver announcement in public transportation: A solution based on speech-to-text and display. In *2020 IEEE 8th International Conference on Smart City and Informatization (iSCI)*, pages 13–19. IEEE.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. 2026. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*.

Khe Chai Sim, Zhouyuan Huo, Tsendsuren Munkhdalai, Nikhil Siddhartha, Adam Stooke, Zhong Meng, Bo Li, and Tara Sainath. 2024. A comparison of parameter-efficient asr domain adaptation methods for universal speech and language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6900–6904. IEEE.

Helmer Strik, Albert Russel, Henk Van Den Heuvel, Catia Cucchiari, and Lou Boves. 1997. A spoken dialog system for the dutch public transport information service. *International Journal of Speech Technology*, 2(2):121–131.

Jiyang Tang, Kwangyoun Kim, Suwon Shon, Felix Wu, and Prashant Sridhar. 2024. Improving asr contextual biasing with guided attention. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12096–12100. IEEE.

A RailKosh Annotation Toolkit

The RailKosh Annotation Toolkit (illustrated in Figure 3) was used to annotate all 1,513 clips

in the RailKosh corpus. The interface provides silver transcript reference options like Google S2T with a one-click *Use this* button to pre-fill the gold transcript editor, significantly reducing manual transcription effort. Hinglish typing support converts Roman input to Devanagari in real time (e.g., *platform* → suggested Devanagari variants shown as chips). Structured fields are presented bilingually (Hindi/English labels) for accessibility to annotators with varying Hindi proficiency. Quality flags (*inaudible/unclear*, *code-mixed*, *needs review*) allow annotators to flag problematic clips for exclusion or secondary review, ensuring corpus quality. The toolkit is designed to be domain-agnostic and language-agnostic, making it directly reusable for extending **RailKosh** to additional Indian languages (Marathi, Tamil, Telugu, Bengali) and new stations in future work. The toolkit is available at <https://github.com/raianand1991/railkosh-toolkit>.

B User Study: Detailed Analysis

Native Language	N (EN)	Task (EN)	Task (HI)
Hindi	189	70.0%	81.6%
English	121	66.7%	87.8%
Bengali	34	44.1%	88.6%
Gujarati	20	36.7%	91.7%
Telugu	10	93.3%	86.7%

Table 6: Task Score by participant’s native language, split by clip language (N = clip responses; groups with $N < 10$ clip responses omitted). Even native English speakers score substantially higher on Hindi clips than English clips, indicating that English clips’ lower scores reflect acoustic degradation in the audio itself rather than language unfamiliarity.

Metric	With	Without	Δ	p
Overall Task Score	74.2%	70.0%	+4.2	0.728
English	65.4%	61.4%	+4.0	0.507
Hindi	84.7%	79.2%	+5.5	0.524
Train No. accuracy	72.9%	64.2%	+8.7	0.138
Platform accuracy	71.2%	71.2%	+0.0	0.719
Type accuracy	80.6%	76.4%	+4.1	0.492

Table 7: With vs without transcript (Mann–Whitney U , per-participant). All comparisons show a positive but non-significant trend favoring *with transcript*, except platform (no difference); none reach $p < 0.05$ at this sample size.

Four findings stand out. First, cross-referencing task score with participants’ native language (Table 6) shows that even native English speakers

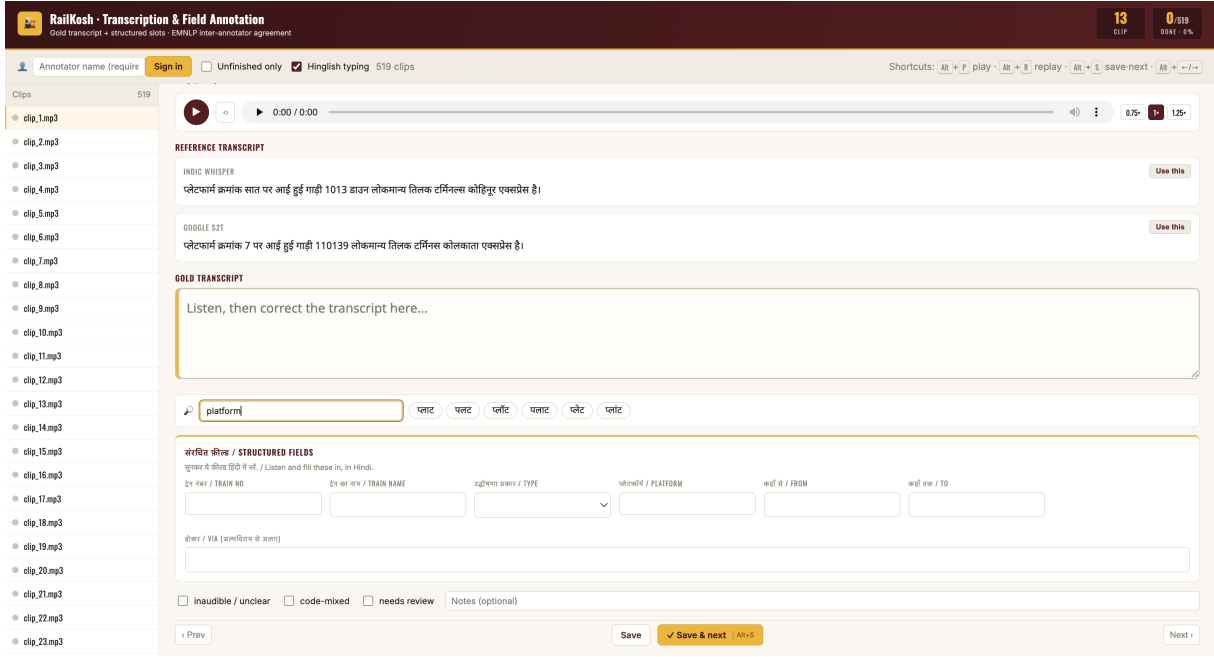


Figure 3: RailKosh Annotation Toolkit interface. The tool displays two silver transcripts (Indic Whisper and Google S2T) as reference, a gold transcript editor with Hinglish typing support, and structured fields (Train No., Train Name, Type, Platform, From) in Hindi. Keyboard shortcuts (Alt+P play, Alt+R replay, Alt+S save-next) accelerate annotation. Progress of each annotator is saved in Google Drive and can be resumed across sessions. Inter-annotator agreement can be calculated at the end.

Group	N	Task Score
<i>Travel Frequency</i>		
Rarely (<1/year)	121	72.7%
Occasionally (1–6/year)	457	74.3%
Regularly ($\geq$ monthly)	194	75.5%
<i>Age Group</i>		
18–24	244	73.2%
25–34	266	76.3%
35–44	136	79.2%
45–54	66	78.8%
55+	60	55.0%

Table 8: Task Score by travel frequency and age group (N = clip responses across all participants who provided demographic data, not distinct participants; each participant contributed up to 10 responses.). The travel-frequency effect is now weak; age effects remain non-monotonic, with the 55+ group scoring lowest.

score markedly higher on Hindi clips than English clips (87.8% vs. 66.7%), and this pattern holds across every native-language group with sufficient data — indicating that English clips’ lower scores reflect acoustic degradation in a subset of the underlying audio rather than a language-comprehension effect, consistent with the near-zero SNR reported for RailKosh (Table 1). Second, transcripts show a positive trend across all metrics (Table 7). Third, age effects are non-monotonic, and the 55+ group

scores notably lower (55.0%) than every other age band; the travel-frequency effect seen in earlier data has also weakened substantially with the larger sample. Fourth, the transcript benefit is concentrated among participants hearing clips in their native language (+8.9 points) rather than their non-native language (−0.4 points); a same-language transcript aids comprehension only when the listener can already read it fluently, whereas closing a genuine language gap requires translation rather than transcription alone. This is precisely why RAILVAANI pairs its transcript output with on-demand translation into eight additional Indian languages, rather than offering transcription alone.