



Analysis of Real-Time Speech Transcription Models

Susmita Sahu¹, Shruti Patel², Nidhi Sinha³, Deepti Chandran⁴

Department of Computer Engineering, Smt. Indira Gandhi College of Engineering, Navi Mumbai, India

ABSTRACT—

Real-time speech transcription turns spoken language into text as it is spoken. It is the basis of live captioning, lecture and meeting notes, dictation and voice interfaces, and it depends on automatic speech recognition (ASR) that stays accurate across languages, accents and mixed-language speech with low delay. This paper analyses forty-one studies published between 2022 and 2026 that bear on building and adapting such models. All were identified through Google Scholar and are grouped into themes: massively multilingual foundation models, Whisper adaptation, parameter-efficient fine-tuning and adapters, streaming architectures, voice activity detection, language identification, speaker diarization, speech translation, and lecture captioning and evaluation. Eighteen studies concern Indian languages and twelve concern other languages or very large multilingual systems. A single comparative table, built only from what the published abstracts state, lists the model, approach, languages and key finding of every study. No single model is best on every measure: the Universal Speech Model, an industrial Conformer/RNN-T system and the Stateful FastConformer stand out for efficiency, while adapters and LoRA stand out for low-cost adaptation. In the reviewed abstracts, spontaneous speech and accuracy under streaming are reported as difficult, and none describes joint handling of language switching, speaker changes and latency. The paper closes with research gaps and future directions, including unified evaluation of accuracy and delay, dialect-aware adaptation and classroom-oriented benchmarks.

Keywords—*automatic speech recognition, fine-tuning, real-time transcription, multilingual ASR, code-switching, Indian languages, Whisper, parameter-efficient adaptation, streaming ASR, speech translation*

I. Introduction

Real-time speech transcription converts speech into text while the speaker is still talking, and it has become one of the most important uses of automatic speech recognition (ASR). ASR is the technology that maps an audio signal to written words. It underpins how people now interact with computers, take notes, learn and access information. When transcription is fast and accurate enough to keep pace with speech, spoken content becomes searchable, shareable and accessible to people who cannot hear it clearly or who speak another language.

Its applications include live captioning for deaf and hard-of-hearing students [1], lecture transcription [2], meeting notes and dictation, on-device voice interfaces [3], [4] and speech translation [5], [6], all of which depend on low word error, low delay and robustness to real speech.

Real speech is hard to handle. Speakers have regional accents and use specialised vocabulary. In many multilingual regions, including India, they also switch between two languages within a single sentence, which is ordinary conversational behaviour rather than an exception [7], [8]. Streaming recognition must also commit to output without seeing future audio, which makes it harder than offline transcription [2], [9].

Large pre-trained speech models have changed how recognisers are built. Instead of training from scratch, practitioners start from a model pre-trained on very large amounts of audio and adapt it to a target language, domain or speaker group with comparatively little labelled data [10]–[12]. The central engineering questions are how to adapt (full fine-tuning, adapters or low-rank updates) and how to combine adaptation with streaming, language identification and speaker handling.

This paper analyses recent work on these questions. It does not describe a particular system but synthesises what forty-one published studies report. Its contributions are (i) a documented literature selection, (ii) a thematic analysis of foundation models, adaptation methods, streaming, supporting components and evaluation, (iii) dedicated treatment of Indian-language, foreign-language and code-switched recognition, (iv) a comparative table of all studies with an assessment of which models perform most efficiently, and (v) research gaps and future directions. The remainder of the paper is organised as follows: Section II describes the methodology, Section III gives background, Section IV reviews the literature, Section V compares the studies, Sections VI and VII discuss gaps and future work, and Section VIII concludes.

II. Review Methodology

A. Sources and search procedure

Candidate papers were identified through searches on Google Scholar, the only index used. Results were limited to 2022–2026 with Scholar's date filter, and only records from the venue categories IEEE Xplore, the ACL Anthology, the ISCA/Interspeech archive, ACM, Springer, Elsevier, MDPI, PeerJ Computer Science, Big Data Mining and Analytics, Nature and AAAI proceedings were accepted; no blogs or general web pages were used. Queries

combined topic terms such as ‘Whisper fine-tuning’, ‘Hindi-English code-switching speech recognition’, ‘Marathi speech recognition’, ‘IndicVoices’, ‘IndicSUPERB’, ‘massively multilingual speech’, ‘parameter-efficient adapters speech recognition’, ‘streaming speech recognition latency’, ‘wav2vec 2.0 domain adaptation’, ‘industrial scale multilingual ASR’, ‘voice activity detection’, ‘spoken language identification’, ‘speaker diarization’, ‘speech translation Indian languages’, ‘lecture transcription classroom’ and ‘real-time captioning’. After de-duplication, distinct candidate records with a DOI or arXiv identifier and an abstract were available for screening.

B. Inclusion and exclusion criteria

A study was included if it (a) was published between 2022 and 2026 inclusive, (b) appeared in a peer-reviewed venue from the list above or, for at most one-fifth of the corpus, was a clearly cited preprint, (c) addressed speech recognition, its adaptation, or a component or evaluation directly relevant to real-time transcription, and (d) had a retrievable abstract. Studies were excluded if they fell outside the year range, could not be matched to a Google Scholar record linking to a DOI or arXiv identifier, duplicated another entry, or concerned unrelated tasks such as emotion recognition, hate-speech text classification or speech synthesis alone.

C. Verification and selection counts

Each shortlisted paper was located in Google Scholar and its publisher or arXiv page was opened to confirm that the record existed, that its year lay in 2022–2026, and that an abstract was retrievable. The final corpus has forty-one studies. Of these, eighteen concern Indian languages (including English-Hindi code-switching), twelve concern foreign languages or very large multilingual models, and eleven address other themes; two are arXiv preprints (4.9% of the corpus). Screening used titles and abstracts only. Full texts were not analysed, so all statements in this review are limited to what the abstracts report; where an abstract does not state a detail, the table records ‘Not reported’.

III. Background

A. The ASR pipeline

A modern ASR system maps an audio waveform to text. The audio is converted to acoustic features, an encoder produces a sequence of representations, and a decoder or output layer turns these into tokens. Output layers have been built with connectionist temporal classification, RNN-transducers and attention-based encoder-decoders; hybrid systems that combine acoustic models with weighted finite-state decoding, as in Kaldi-based work [13], remain in use. Quality is normally reported as word error rate (WER) or character error rate (CER); code-switching studies additionally use mixed error rate (MER) and transliterated WER [8], [14].

B. Real-time and streaming recognition

Real-time note transcription requires emitting text while the speaker is still talking. Streaming models therefore limit how much future audio the encoder may see and cache past computation so that each new audio chunk is processed once [9]. The latency of the whole path, from microphone to displayed text, matters as much as accuracy; hardware-oriented work reports end-to-end delay and real-time factor explicitly [3]. Components such as voice activity detection decide when to process audio at all and must themselves run within tight latency budgets [4].

C. Fine-tuning approaches

Pre-trained models are adapted in several ways. Full fine-tuning updates all weights and has been explored in depth for Whisper under limited data [10]. Parameter-efficient methods update only a small fraction of the model: adapters inserted into wav2vec 2.0 allow tasks or languages to be added while training under a tenth of the parameters [12], and low-rank adaptation (LoRA) was applied to Whisper for Turkish [15]. Domain adaptation of foundation models can also use unlabelled target-domain data together with a small supervised set [16]. Self-supervised pre-training on unlabelled speech, followed by supervised fine-tuning, underlies much of the multilingual and Indic work reviewed below [17], [18].

IV. Literature Review

A. Massively multilingual foundation models

The Universal Speech Model pre-trains an encoder on an unlabelled corpus spanning more than 300 languages and then fine-tunes on smaller labelled data to recognise over 100 languages; its authors report comparable or better results than Whisper despite using about one-seventh of the labelled data [11]. SeamlessM4T unifies speech-to-speech, speech-to-text, text-to-speech, text-to-text translation and ASR in one model covering up to 101 input languages, and is reported to outperform cascaded systems in BLEU and to be more robust to noise and speaker variation [19]. Because Whisper’s training pipeline is not public, the Open Whisper-style Speech Model reproduces Whisper-style training with an open toolkit and public data and releases its scripts and models [20]. A study on 70 languages compares a shared and a multiple-embedding design, stresses the importance of tokenisation, and reports average relative WER gains over monolingual models together with zero-shot and fine-tuned results on an unseen dataset [21]. Work on 15 low-resource languages from the OpenASR21 challenge analyses wav2vec 2.0, HuBERT and WavLM models, fine-tuning variants and layer-wise representation similarity [18]. An industrial system from AssemblyAI pairs a 600M-parameter full-context Conformer encoder, pre-trained with BEST-RQ on 12.5M hours of unlabelled audio, with an RNN-T decoder fine-tuned jointly with the encoder; trained on 188k hours of supervised and 1.6M hours of pseudo-labelled data across four languages, it is reported by its authors to reach word error rates competitive with larger and more expensive models such as Whisper large and Canary-1B [22].

B. Whisper adaptation and fine-tuning strategies

Several studies treat Whisper as the starting point for low-resource recognition. A systematic comparison of five fine-tuning strategies on seven low-resource languages found that all of them help, but that their suitability depends on use case and available resources [10]. For Arabic, fine-tuning Whisper

on Modern Standard Arabic data gave substantial gains for smaller models, while pooling dialect data performed comparably to dialect-specific models [23]. Fine-tuning Whisper Tiny for Swahili improved accuracy but left persistent errors on phonetic confusions, named entities and morphologically complex words [24]. A Turkish system applied LoRA to Whisper and reported large WER reductions over the baseline [15]. For Indian-accented English, fine-tuned Whisper Tiny, Small and Medium models were evaluated on a newly collected dataset of about 170 hours [25], and a web application packages Whisper for offline Indic speech-to-text and translation [26].

C. Parameter-efficient adaptation and domain adaptation

Adapters were applied to wav2vec 2.0 so that downstream ASR could be trained with fewer than 10% of the parameters of full fine-tuning at little cost in accuracy, and placing adapters only in the top layers performed similarly [12]. In a large-scale foundation-model setting, pre-training with BEST-RQ and fine-tuning an encoder adapter and decoder on small in-domain data matched the quality of a much larger model trained from scratch while using far fewer supervised examples and trainable parameters [16]. ELP-adapter tuning combines encoder, layer and prompt adapters for parameter-efficient fine-tuning of self-supervised speech models across several tasks; with a WavLM backbone it was comparable to or better than full fine-tuning while requiring 90% fewer learnable parameters [27]. Together with the Whisper LoRA study [15], these works suggest that adaptation need not touch most of a model's weights, a property that matters when many languages or users must each hold their own adapted copy.

D. Streaming and low-latency recognition

The Stateful Conformer adapts FastConformer to streaming by constraining look-ahead and past context and caching activations so that a non-autoregressive encoder can run incrementally, removing the usual mismatch between training and inference and giving better accuracy at lower latency than a buffered baseline [9]. At the other end of the stack, a streaming on-device system pairs an FPGA acoustic model with a smartphone language model and reports a worst-case latency of 125.5 ms and a real-time factor of 0.57 [3]. A Whisper-based system for multilingual, multi-speaker settings integrates diarization and is optimised for real-time use [28]. An evaluation of 11 common ASR services on lecture recordings reports significantly lower quality for streaming ASR than for other modes [2].

E. Voice activity detection, language identification and diarization

Personal VAD 2.0 detects the voice activity of a target speaker within a streaming on-device ASR system, using speaker-embedding modulation, training for enrolment-less conditions and runtime optimisation to meet latency and memory limits [4]. For spoken language identification, using wav2vec 2.0 features with an x-vector self-attention model, plus squeeze-and-excitation and linear-bottleneck blocks to discard irrelevant information, gave a 63.79% relative improvement in average cost on AP19-OLR data when wav2vec 2.0 features replaced the earlier input features, while the added blocks gave a separate, smaller gain of 9.85% on AP19-OLR data and over 10% overall on MLS14 data with a modest number of extra parameters [29]. The concatenated-tokenizer approach also emits a language label per token and reaches over 98% language-identification accuracy on FLEURS [30]. pyannote.audio 2.1 is an open-source Python toolkit whose diarization pipeline has three stages: speaker segmentation on a short sliding window, neural embedding of the local speakers, and global agglomerative clustering; it also includes a recipe for improving performance on a user's own annotated data, and it is reported to have placed first at Ego4D 2022, first at Albayzin 2022 and sixth at VoxSRC 2022 [31]. A Whisper-based system combining recognition with speaker embeddings reports word diarization error rates for two-speaker, three-speaker and overall conditions on Mandarin with Taiwanese accents [28].

F. Speech translation: cascaded versus end-to-end

For Indian languages, the NICT submission to IWSLT 2024 compared a cascade of fine-tuned Whisper and IndicTrans2 against an end-to-end Whisper model trained with distilled IndicTrans2 knowledge, and found the cascade far ahead in BLEU, while distillation still helped the end-to-end model [5]. A large dataset of over 44 thousand hours covering 14 Indian languages and English was used to train Indic-Seamless, which is reported to beat existing models [32]. A configurable speech-to-speech pipeline for English-to-Indian-language video translation chains ASR, text pre-processing, machine translation, text-to-speech and rendering, with human editing between modules [6]. SeamlessM4T takes the opposite route of a single unified model [19]. The evidence therefore does not settle the cascaded-versus-end-to-end question; it depends on language pair, data and the quality of the component models.

G. Lecture and classroom captioning and evaluation

Evidence specific to classrooms is limited in the reviewed set. A prototype combining real-time captioning and live transcription for deaf and hard-of-hearing students was assessed through in-class observation and showed, according to its authors, potential to help students follow lectures and class discussions [1]. A broader measurement of 11 common ASR services on higher-education lectures reported wide variation in accuracy between vendors and audio samples and lower streaming quality, concluding that common services lack reliability in accuracy [2]. Other studies report WER, CER, MER, BLEU and latency measures such as real-time factor [3], [5], [8], [14], but none of the reviewed abstracts describes a unified benchmark that scores accuracy and delay together for lecture speech.

H. Multilingual and code-switched recognition

1) *Indian languages.* Dataset and benchmark work provides the foundation. IndicVoices collects 7,348 hours of read, extempore and conversational speech from 16,237 speakers in 145 districts across 22 languages, and trains IndicASR, the first model covering all 22 scheduled languages [33]. IndicSUPERB contributes the 1,684-hour Kathbath corpus for 12 languages and benchmarks self-supervised models on six speech tasks [34]. All Ears pre-trains five SSL architectures on about 29.5K hours across 24 languages and reports relative WER gains over MMS-300M and IndicWav2Vec [17]. MADASR 2.0 provides over 1,200 hours across 8 languages and 33 dialects and finds spontaneous speech still difficult while multitask and transfer learning help [35]. Language-specific efforts include Tamil recognition with XLSR wav2vec 2.0 and CTC on Common Voice [36], a Seq2Seq Tamil-English system [37], a Kaldi-based Marathi system trained on about 95 hours [13], a wavelet-transformer model for Indian languages [38], and surveys of Marathi and Bengali recognition [39], [40]. Indian-accented English has its own dataset and Whisper fine-tuning study [25], and a concatenated-tokenizer model covers English-Hindi code-switching [30].

2) *Foreign languages*. Beyond the foundation models above, fine-tuning studies cover Arabic dialects [23], Swahili [24], Turkish [15] and seven further low-resource languages [10]. Large multilingual systems cover 70 languages [21], 100 or more languages [11], [19], 15 OpenASR21 languages [18] and four languages in an industrial system [22]. Mandarin with Taiwanese accents is addressed in the real-time diarization system [28].

3) *Code-switching*. For Hindi-English, a transformer-transducer whose language-specific modules are pre-trained on monolingual data reached a WER of 29.63% and a transliterated WER of 27.42% [8], while a Whisper adaptation using language prompts, on-the-fly simulated code-mixing and Llama-generated text with synthetic audio showed promising results [7]. A concatenated-tokenizer model creates code-switched training data from monolingual sources and sets new results on the Miami Bangor corpus for English-Hindi and English-Spanish [30]. For Mandarin-English, cyclic transfer learning alternating between code-switched and monolingual tasks reached 16.3% and 24.1% MER on the two SEAME test sets [14], and a mutual-learning semi-supervised method using pseudo-labelled data lowered these to 15.6% and 21.1% [41]. Common threads are scarcity of code-switched speech and the use of monolingual or synthetic data to compensate.

V. Comparative Table

Table I lists every reviewed study. Entries are taken from the abstracts; ‘Not reported’ means the abstract does not state the item; ‘n/a’ means not applicable.

TABLE I
Comparative Summary of the Reviewed Studies (from Published Abstracts)

Ref.	Year	Model	Approach	Language(s)	Key finding
[1]	2023	Real-time captioning prototype	Captioning and live transcription; classroom observation	Not reported	Authors report potential to help DHH students follow lectures
[2]	2023	11 common ASR services	Accuracy measurement on higher-education lecture recordings	Not reported	Wide variance by vendor and sample; significantly lower streaming quality
[3]	2022	CNN acoustic model on FPGA + 3-gram LM	On-device streaming, 80 ms input	English (LibriSpeech test-clean)	Worst-case latency 125.5 ms; RTF 0.57; WER 13.2% (acoustic model), 9.1% (with language model)
[4]	2022	Personal VAD 2.0	Speaker-embedding modulation, enrolment-less training	Not reported	Best reported performance on a realistic ASR system
[5]	2024	Whisper + IndicTrans2	Cascade vs. end-to-end with knowledge distillation	English to Hindi, Bengali, Tamil	Cascade 51.0 BLEU vs. 19.1 end-to-end; distillation +0.7 BLEU
[6]	2023	Cascaded ASR-MT-TTS pipeline	Human-in-the-loop speech-to-speech video translation	English to Hindi, Telugu, Gujarati, Marathi, Punjabi	MOS 3.5+ for English-Hindi with human intervention
[7]	2025	Whisper	Language prompts, on-the-fly code-mix simulation, Llama text + synthetic audio	Hindi-English	Promising results on Hindi-English CS data
[8]	2022	Transformer-transducer	Language-specific modules pre-trained on monolingual data	Hindi-English	WER 29.63%, T-WER 27.42%; 2.19% better than prior best
[9]	2024	Stateful FastConformer	Limited context, activation caching; CTC/RNNT	Not reported	Better accuracy, lower latency than buffered baseline
[10]	2024	Whisper	Five fine-tuning strategies	Seven low-resource languages	All strategies help; suitability varies
[11]	2023	USM	Pre-train on 12M h, 300+ languages; fine-tune	100+ languages	Comparable or better than Whisper with 1/7 labelled data
[12]	2022	wav2vec 2.0 + adapters	Adapters for downstream ASR	Not reported	<10% of parameters trained with little degradation
[13]	2023	Kaldi GMM-HMM / DNN-HMM	LVCSR on ~95 h with trigram LM	Marathi	Average WER 19%
[14]	2023	Cyclic transfer learning	Alternate CS and monolingual pretext tasks	Mandarin-English	MER 16.3% / 24.1% (SEAME test sets)
[15]	2024	Whisper	LoRA fine-tuning	Turkish	WER reduced by up to 52.38%
[16]	2023	BEST-RQ + JUST Hydra	Encoder adapter + decoder fine-tuning	Not reported	Same quality with 21.6M in-domain data, 130.8M tuned params vs. 731.1M model
[17]	2024	SSL models, 5 architectures	Pre-train on ~29.5K h, fine-tune for ASR	24 (pre-train), 13 (ASR) Indian languages	17.3% and 36.0% relative WER gain over MMS-300M and IndicWav2Vec
[18]	2022	wav2vec 2.0, HuBERT, WavLM	Pre-training and fine-tuning variants; layer analysis	15 OpenASR21 languages	Better results in E2E and hybrid systems (no figure given)

Ref.	Year	Model	Approach	Language(s)	Key finding
[19]	2025	SeamlessM4T	Single multitask multimodal model	Up to 101 input languages	Up to 8% / 23% higher BLEU than cascades; ~50% more noise-robust
[20]	2023	OWSM	Whisper-style training with open toolkit and data	Multilingual (count not reported)	More translation directions; more efficient training
[21]	2023	Multilingual E2E (shared / multiple embeddings)	Tokenisation study across languages	70 languages	13.9-15.6% avg relative WER gain; 9.5% / 7.5% WER on MLS
[22]	2024	Conformer (600M) + RNN-T	BEST-RQ pre-training on 12.5M h; 188k h supervised plus 1.6M h pseudo-labelled; joint fine-tuning	Four languages	Competitive WER vs. larger Whisper large and Canary-1B
[23]	2025	Whisper	Fine-tune on MSA and dialect data	Arabic (5 dialects)	Small MSA data helps smaller models; pooled ~ dialect-specific
[24]	2025	Whisper Tiny (39M)	Fine-tune on 5,520 samples	Swahili	Gains, but errors persist on phonetics, entities, morphology
[25]	2025	Whisper Tiny/Small/Medium	Fine-tuning on ~170 h newly collected data	Indian-accented English	WER 18.141% (Tiny), 17.36% (Small), 15.08% (Medium)
[26]	2025	Whisper	Offline web app for ASR and translation	Telugu, Hindi, English	Not reported
[27]	2024	ELP-adapter tuning (E-, L- and P-adapters); five backbone models	Parameter-efficient adapter tuning of self-supervised models; evaluated on four downstream tasks	Not reported	With WavLM, comparable to or better than full fine-tuning with 90% fewer learnable parameters
[28]	2024	Whisper + speaker embeddings	Real-time ASR with diarization	Mandarin (Taiwanese accents)	WDER 2.68% (2 speakers), 11.65% (3), 6.96% overall
[29]	2022	wav2vec 2.0 + XSA-LID	wav2vec 2.0 input features; SE and linear-bottleneck (LBN) blocks	Not reported	63.79% relative gain in average cost (AP19-OLR) from wav2vec 2.0 features; LBN-XSA adds 9.85% on AP19-OLR
[30]	2023	Concatenated-tokenizer ASR	CS data built from monolingual sources; per-token language ID	English-Hindi, English-Spanish	New best on Miami Bangor; 98%+ LID accuracy on FLEURS
[31]	2023	pyannote.audio 2.1	Open-source toolkit; 3-stage pipeline (sliding-window speaker segmentation, neural embedding of local speakers, global agglomerative clustering); recipe for tuning on own annotated data	Not reported	1st at Ego4D 2022, 1st at Albayzin 2022, 6th at VoxSRC 2022
[32]	2025	Indic-Seamless	BhasaAnuvaad dataset from aggregation, crawling and synthetic disfluent data	14 Indian languages + English	Reported to outperform existing models in translation quality
[33]	2024	IndicASR	Large-scale spontaneous-speech collection with open blueprint, then ASR training	22 Indian languages	7,348 h from 16,237 speakers (1,639 h transcribed); first ASR covering all 22 scheduled languages
[34]	2023	Self-supervised models (several)	Kathbath corpus; benchmarks on 6 speech tasks	12 Indian languages	1,684 h labelled speech; benchmark for ASR, speaker, LID, QbE and keyword tasks
[35]	2025	Not reported	Challenge on 1,200+ h read speech (RESPIN subset); 4 tracks; auxiliary language/dialect ID tasks	8 Indian languages, 33 dialects	Spontaneous ASR still hard; multitask and transfer learning help
[36]	2022	XLSR Wav2Vec 2.0	Cross-lingual transfer + CTC	Tamil	WER 58%, CER 11%
[37]	2024	Seq2Seq E2E	End-to-end framework with language segmentation	Tamil-English	94% accuracy (metric unspecified)
[38]	2023	Wavelet transformer (WTASR)	Wavelet decomposition fed to transformer	Indian language (unspecified)	Low WER vs. compared methods (no figure given)
[39]	2022	n/a (survey)	Review of Marathi recognition and techniques	Marathi	Not reported
[40]	2023	n/a (survey)	Review of models 2009-2022, challenges, datasets	Bengali	Lists models, datasets and future scope

Ref.	Year	Model	Approach	Language(s)	Key finding
[41]	2025	MLSS	Two networks, mutual semi-supervised learning	Mandarin-English	MER 15.6% / 21.1% (SEAME test sets)

VI. Research Gaps

Spontaneous and dialectal speech. In the reviewed abstracts, spontaneous speech stayed difficult for Indian languages even in a dedicated challenge [35], and Arabic dialect work reports minimal benefit from Modern Standard Arabic pre-training for dialects [23].

Accuracy under streaming. In the reviewed abstracts, streaming-mode ASR scored lower than other modes in a lecture evaluation [2], and none of the reviewed abstracts describes an Indian or code-switched evaluation of the streaming architectures [3], [9].

Joint treatment of components. In the reviewed abstracts, code-switching, diarization, VAD, language identification and parameter-efficient adaptation are studied largely in isolation; none of the reviewed abstracts describes a single low-latency pipeline that combines them for note taking.

Code-switched data scarcity. In the reviewed abstracts, code-switching work uses monolingual, synthetic or pseudo-labelled data [7], [8], [30], [41], where the abstracts state data scarcity as the motivation [8], [41], and Indian code-switching is identified in the reviewed abstracts for Hindi-English and Tamil-English [7], [8], [30], [37].

Reporting practice. In the reviewed abstracts, several give no quantitative result or language details, for example [12], [18], [26], [38], which hinders direct comparison.

VII. Future Directions

Lightweight per-language or per-domain adaptation, using adapters or low-rank updates [12], [15], [16], applied to Indic and code-switched speech and evaluated in streaming mode.

Data strategies that combine monolingual pre-training, synthetic code-switched text and audio, and pseudo-labelling [7], [8], [41], validated on spontaneous speech.

Streaming-aware multilingual models that carry language identification and speaker information per token or segment [9], [28], [30].

Larger Indic resources and benchmarks [33]–[35] extended with lecture and meeting recordings, with protocols that report WER, latency and real-time factor together.

Rigorous classroom studies with end users, building on early captioning prototypes [1] and on independent accuracy measurement [2].

Clearer comparison of cascaded and end-to-end routes when translation is added to transcription [5], [19], [32].

VIII. Conclusion

This paper analysed forty-one Google Scholar-indexed studies from 2022 to 2026 on real-time, multilingual and code-switched speech transcription. Several studies report that adapting large pre-trained models to a target language with limited data improves recognition, and that adaptation can be made parameter-efficient. They also show that Indian-language recognition now has large datasets and benchmarks, and that code-switched recognition benefits from monolingual and synthetic data.

No model is best on every measure, because the studies use different tasks, languages and metrics; within each criterion, some stand out. For streaming efficiency, the Stateful FastConformer gives better accuracy at lower latency than a buffered baseline [9], and an FPGA on-device system reports 125.5 ms worst-case latency with a real-time factor of 0.57 [3]. For accuracy per unit of cost, the Universal Speech Model matches or beats Whisper with about one-seventh of the labelled data [11], and AssemblyAI's Conformer/RNN-T system reports competitive WER against larger models, although its encoder is full-context rather than streaming and its results are self-reported [22]. For adaptation efficiency, adapters reach near-full-fine-tuning accuracy while training under 10% of parameters [12], ELP-adapters need 90% fewer learnable parameters with a WavLM backbone [27], and LoRA reduced Turkish Whisper WER by up to 52.38% [15]. For code-switching, the mutual-learning MLSS method lowered Mandarin-English MER to 15.6% and 21.1% on the two SEAME test sets [41].

For a real-time note-taking system, the evidence favours a streaming Conformer-type encoder with lightweight adapters, although the reviewed abstracts do not show this combination evaluated on Indian or code-switched speech. They also point to remaining gaps in spontaneous speech, streaming accuracy and integrated pipelines. Because the synthesis rests on abstracts alone, readers should consult the cited full texts before relying on any specific figure.

Disclosure: AI tools were used to assist literature retrieval and drafting; the authors reviewed the content.

REFERENCES

- [1] S. Pal, V. Badhe, and C. Dasgupta, "Investigating the role of closed captioning and live transcription on DHH students' perception of inclusivity in a face-to-face classroom environment," in *Proc. IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, 2023, doi: 10.1109/icalt58122.2023.00070.
- [2] K. Kuhn et al., "Measuring the accuracy of automatic speech recognition solutions," *ACM Trans. Access. Comput.*, 2023, doi: 10.1145/3636513.
- [3] J. Park et al., "A low-latency streaming on-device automatic speech recognition system using a CNN acoustic model on FPGA and a language model on smartphone," *Electronics*, 2022, doi: 10.3390/electronics11121831.
- [4] S. Ding et al., "Personal VAD 2.0: Optimizing personal voice activity detection for on-device speech recognition," in *Proc. Interspeech*, 2022, doi: 10.21437/interspeech.2022-856.
- [5] R. Dabre and H. Song, "NICT's cascaded and end-to-end speech translation systems using Whisper and IndicTrans2 for the Indic task," in *Proc. 21st Int. Conf. Spoken Lang. Transl. (IWSLT)*, 2024, doi: 10.18653/v1/2024.iwslt-1.3.
- [6] V. Mujadia and D. Sharma, "Towards speech to speech machine translation focusing on Indian languages," in *Proc. 17th Conf. Eur. Chapter Assoc. Comput. Linguistics: Syst. Demonstrations*, 2023, doi: 10.18653/v1/2023.eacl-demo.19.

- [7] A. Biswas et al., "Adapting Whisper for low-resource Hindi-English code-mix speech with on-the-fly augmentation & LLM-synthesised data," in *Proc. Interspeech*, 2025, doi: 10.21437/interspeech.2025-447.
- [8] A. Antony et al., "An improved transformer transducer architecture for Hindi-English code switched speech recognition," in *Proc. Interspeech*, 2022, doi: 10.21437/interspeech.2022-10763.
- [9] V. Noroozi et al., "Stateful Conformer with cache-based inference for streaming automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2024, doi: 10.1109/icassp48485.2024.10446861.
- [10] Y. Liu, X. Yang, and D. Qu, "Exploration of Whisper fine-tuning strategies for low-resource ASR," *EURASIP J. Audio, Speech, Music Process.*, 2024, doi: 10.1186/s13636-024-00349-3.
- [11] Y. Zhang et al., "Google USM: Scaling automatic speech recognition beyond 100 languages," 2023, arXiv:2303.01037.
- [12] B. Thomas, S. Kessler, and S. Karout, "Efficient adapter transfer of self-supervised speech models for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, doi: 10.1109/icassp43922.2022.9746223.
- [13] S. C. Bajaj et al., "Large vocabulary continuous speech recognition system for Marathi," in *Proc. Int. Conf. Adv. Electron., Commun., Comput. Intell. Inf. Syst. (ICAECIS)*, 2023, doi: 10.1109/icaecis58353.2023.10170653.
- [14] C. H. Nga et al., "Cyclic transfer learning for Mandarin-English code-switching speech recognition," *IEEE Signal Process. Lett.*, 2023, doi: 10.1109/LSP.2023.3307350.
- [15] H. Polat et al., "Implementation of a Whisper architecture-based Turkish automatic speech recognition (ASR) system and evaluation of the effect of fine-tuning with a low-rank adaptation (LoRA) adapter on its performance," *Electronics*, 2024, doi: 10.3390/electronics13214227.
- [16] B. Li et al., "Efficient domain adaptation for speech foundation models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, doi: 10.1109/icassp49357.2023.10096330.
- [17] V. S. Lodagala et al., "All Ears: Building self-supervised learning based ASR models for Indian languages at scale," in *Proc. Interspeech*, 2024, doi: 10.21437/interspeech.2024-2520.
- [18] J. Zhao and W.-Q. Zhang, "Improving automatic speech recognition performance for low-resource languages with self-supervised models," *IEEE J. Sel. Topics Signal Process.*, 2022, doi: 10.1109/jstsp.2022.3184480.
- [19] SEAMLESS Communication Team, L. Barrault et al., "Joint speech and text machine translation for up to 100 languages," *Nature*, 2025, doi: 10.1038/s41586-024-08359-z.
- [20] Y. Peng et al., "Reproducing Whisper-style training using an open-source toolkit and publicly available data," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2023, doi: 10.1109/asru57964.2023.10389676.
- [21] A. Tjandra et al., "Massively multilingual ASR on 70 languages: Tokenization, architecture, and generalization capabilities," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, doi: 10.1109/icassp49357.2023.10094667.
- [22] F. McCann Ramirez et al., "Anatomy of industrial scale multilingual ASR," 2024, arXiv:2404.09841, doi: 10.48550/arXiv.2404.09841.
- [23] Ö. T. Özyilmaz, M. Coler, and M. Valdenegro-Toro, "Overcoming data scarcity in multi-dialectal Arabic ASR via Whisper fine-tuning," in *Proc. Interspeech*, 2025, doi: 10.21437/Interspeech.2025-2260.
- [24] A. K. Sharma, M. Pandya, and A. Shukla, "Fine-tuning Whisper Tiny for Swahili ASR: Challenges and recommendations for low-resource speech recognition," in *Proc. 6th Workshop Afr. Natural Lang. Process. (AfricaNLP)*, 2025, doi: 10.18653/v1/2025.africanlp-1.11.
- [25] J. Kunistetty et al., "Advancing ASR for Indian-accented English: Dataset creation and Whisper fine-tuning," *Procedia Comput. Sci.*, 2025, doi: 10.1016/j.procs.2025.04.513.
- [26] K. Swapna et al., "Multilingual ASR for Indic languages using OpenAI's Whisper," in *Proc. 2nd Int. Conf. Circuits, Power Intell. Syst. (CCPIS)*, 2025, doi: 10.1109/ccpis65231.2025.11234204.
- [27] N. Inoue et al., "ELP-adapters: Parameter efficient adapter tuning for various speech processing tasks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2024, doi: 10.1109/taslp.2024.3434445.
- [28] K.-M. Lyu, R.-y. Lyu, and H.-T. Chang, "Real-time multilingual speech recognition and speaker diarization system based on Whisper segmentation," *PeerJ Comput. Sci.*, 2024, doi: 10.7717/peerj-cs.1973.
- [29] H. Liu et al., "Efficient self-supervised learning representations for spoken language identification," *IEEE J. Sel. Topics Signal Process.*, 2022, doi: 10.1109/jstsp.2022.3201445.
- [30] K. Dhawan, D. Rekish, and B. Ginsburg, "Unified model for code-switching speech recognition and language identification based on concatenated tokenizer," in *Proc. 6th Workshop Comput. Approaches Linguistic Code-Switching*, 2023, doi: 10.18653/v1/2023.calcs-1.7.
- [31] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: Principle, benchmark, and recipe," in *Proc. Interspeech*, 2023, doi: 10.21437/interspeech.2023-105.
- [32] A. Sankar et al., "Towards building large scale datasets and state-of-the-art automatic speech translation systems for 14 Indian languages," in *Proc. 63rd Annu. Meeting Assoc. Comput. Linguistics (Vol. 1: Long Papers)*, 2025, doi: 10.18653/v1/2025.acl-long.1582.
- [33] T. Javed et al., "IndicVoices: Towards building an inclusive multilingual speech dataset for Indian languages," in *Findings Assoc. Comput. Linguistics: ACL 2024*, 2024, doi: 10.18653/v1/2024.findings-acl.639.
- [34] T. Javed et al., "IndicSUPERB: A speech processing universal performance benchmark for Indian languages," in *Proc. AAAI Conf. Artif. Intell.*, 2023, doi: 10.1609/aaai.v37i1.26521.
- [35] S. Kumar et al., "MADASR 2.0: Multi-lingual multi-dialect ASR challenge in 8 Indian languages," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2025, doi: 10.1109/ASRU65441.2025.11434729.
- [36] A. Akhilesh et al., "Tamil speech recognition using XLSR Wav2Vec2.0 & CTC algorithm," in *Proc. 13th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2022, doi: 10.1109/icccnt54827.2022.9984422.
- [37] R. G. Rajakumari et al., "Multilingual automatic speech recognition for Indian languages-E2E framework," in *Proc. OITS Int. Conf. Inf. Technol. (OCIT)*, 2024, doi: 10.1109/ocit65031.2024.00033.
- [38] T. Choudhary, V. Goyal, and A. Bansal, "WTASR: Wavelet transformer for automatic speech recognition of Indian languages," *Big Data Mining Anal.*, 2023, doi: 10.26599/bdma.2022.9020017.
- [39] S. Balshankar and R. R. Deshmukh, "A critical insight into Marathi speech recognition and techniques," in *Proc. 2nd Int. Conf. Artif. Intell. Smart Energy (ICAIS)*, 2022, doi: 10.1109/icais53314.2022.9742925.
- [40] N. Tasnia et al., "An overview of Bengali speech recognition: Methods, challenges, and future direction," in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2023, doi: 10.1109/ccwc57344.2023.10099382.
- [41] C. H. Nga et al., "MLSS: Mandarin English code-switching speech recognition via mutual learning-based semi-supervised method," *IEEE Signal Process. Lett.*, 2025, doi: 10.1109/LSP.2025.3540953.