Bengali-English and Hindi-English Code Mixed Speech Data with Disfluencies

Anuran Mitra*, **Tapabrata Mondal***, **Anirvan Chakravarty***, **Sivaji Bandyopadhyay***
*Jadavpur University, Kolkata, India
{anuranm3, tapabratamondal, anirvanchakravarty39, sivaji.cse.ju}@gmail.com

Abstract

Spontaneous speech in multilingual communities such as India frequently combines code-switching (CS) and disfluencies, yet existing Bengali–English and Hindi–English speech corpora largely consist of fluent or scripted utterances. This limits their suitability for developing and evaluating automatic speech recognition (ASR) systems intended for real conversational settings, particularly in micro-resource scenarios. We introduce BEHE-CMDisfl, a synthetic speech corpus that explicitly integrates disfluency phenomena within Bengali–English and Hindi–English code-mixed (CM) utterances. The textual content was generated using prompting strategies with large language models (LLMs) to encourage controlled switching and varied disfluency patterns, including filled pauses, repetitions, and restarts. The utterances were subsequently synthesized using Indic Parler text-to-speech (TTS) system. To demonstrate usability, we establish a reproducible GMM–HMM baseline for Bengali–English ASR using Kaldi on a 1.3-hour subset of the corpus. In our experiments, improvements were mainly observed after ensuring consistency in the pronunciation lexicon and applying phonetic normalization, with the best setup reaching a word error rate (WER) of 37.74%. A closer look at the decoded transcripts suggests that filled pauses and repetitions are not automatically collapsed, but appear in the output, indicating that the disfluency cues present in the synthetic speech are captured during recognition.

Keywords: code-switching (CS), disfluencies, automatic speech recognition (ASR), code-mixing (CM), large language models (LLMs), text to speech (TTS), word error rate (WER)

1. Introduction

Multilingual communication is a defining feature of the South Asian linguistic landscape. In daily communication, speakers frequently blend languages such as Bengali, Hindi, and English within a single utterance, a phenomenon commonly referred to as code-mixing or code-switching (Auer, 2013; Moyer, 2002). ***Code-mixing (CM)*** refers to the mixing of linguistic units such as morphemes, words, phrases, or clauses from two grammatical systems within a sentence, whereas ***code-switching (CS)*** denotes alternation across sentences or discourse segments. CM is intra-sentential, whereas CS is inter-sentential, but quite often both are used interchangeably (Kim, 2006). This linguistic blending is particularly prevalent in multilingual societies such as India (Harya, 2018), where conversational speech naturally exhibits both cross-lingual alternation and spontaneous irregularities.

Alongside multilingual blending, spontaneous speech exhibits disfluencies such as filled pauses ("uh," "um"), repetitions, hesitations, and restarts, which are natural by-products of human planning and self-monitoring during speech production (Shriberg, 2001; Adell et al., 2006). However, most existing speech corpora for Indic languages consist of scripted, fluent recordings (Saranya et al., 2025; Diwan et al., 2021), and contemporary ASR systems often treat non-lexical tokens such as filled pauses as noise to be removed (Kundu et al., 2022; Zayats et al., 2016). This sanitization strategy simplifies decoding but limits the applicability of ASR systems for conversational AI, sentiment analysis, and speech-based behavioral modeling, where disfluencies carry meaningful information. While several text-based code-mixed resources exist, such as BnSentMix (Alam et al., 2024) and SentMix-3L (Nishat Raihan et al., 2023) but they do not capture acoustic variability. Similarly, studies focused on disfluencies have advanced toward multilingual or synthetic augmentation settings (Bhat et al., 2023; Romana et al., 2024; Amann et al., 2024), yet remain largely English-centric or text-bound. Speech-level corpora that jointly represent code-mixing and disfluency for Indic languages remain scarce. Even in Hindi-English ASR (Sitaram et al., 2019), Bengali-English disfluent speech has received limited attention.

Collecting naturalistic disfluent code-mixed speech is costly and difficult, especially in micro-resource scenarios (approximately one hour of data). Large-scale end-to-end ASR models require hundreds of hours to generalize effectively (Hannun et al., 2014; Watanabe et al., 2018), rendering them unsuitable for cold-start settings. In contrast, Gaussian Mixture Model–Hidden Markov Model (GMM-HMM) systems remain practical and interpretable for severely data-constrained environments (Besacier et al., 2014; Mohri et al., 2008), and the Kaldi toolkit

Proceedings of WILDRE-8, pages 39–48
12 May 2026.

provides a robust experimental framework for such regimes (Povey et al., 2011). Recent advances in Large Language Models (LLMs) and multilingual neural text-to-speech (TTS) systems such as Indic Parler TTS (Lacombe et al., 2024; Lyth and King, 2024) offer an alternative pathway. Synthetic speech generation has been shown to support low-resource ASR research when real data collection is infeasible (Yeo et al., 2026). Rather than aiming to replace natural corpora, synthetic pipelines provide a controlled environment for modeling specific linguistic phenomena. Building on these advances, we construct **BEHE-CMDisfl** by generating disfluent Bengali–English and Hindi–English code-mixed utterances using structured LLM prompts using OpenAI's **ChatGPT**[1], followed by speech synthesis with **Indic Parler TTS**[2]. The corpus explicitly incorporates filled pauses, repetitions, and restarts within multilingual speech. Beyond resource creation, we evaluate its practical utility in a micro-resource setting by establishing GMM-HMM baselines on a 1.3-hour Bengali–English subset using Kaldi. This study therefore pursues two complementary goals: to release a disfluency-aware code-mixed speech corpus, and to examine its suitability for low-resource ASR experimentation.

1.1. Our Contribution:

- We develop **BEHE-CMDisfl**, a Bengali–English (BE) and Hindi–English (HE) speech dataset generated through a controlled LLM–TTS pipeline that explicitly integrates filled pauses, repetitions, and restarts within multilingual utterances.
- We document the prompt design strategy, linguistic constraints, and synthesis pipeline to ensure reproducibility and clarity in the data creation process.
- We establish reproducible GMM-HMM baselines (Monophone, Triphone, LDA-MLLT) for Bengali–English disfluent code-mixed ASR using Kaldi on a 1.3-hour subset.
- We demonstrate the impact of transliteration normalization and lexicon consistency on recognition performance, achieving a best WER of 37.74%.
- We show that disfluency markers are retained in decoding outputs when explicitly modeled in the lexicon, supporting their inclusion rather than removal in conversational ASR research.

[1] https://chatgpt.com/

[2] https://huggingface.co/ai4bharat/indic-parler-tts

2. Related Works

Research on code-mixed language processing has expanded in recent years, particularly for multilingual contexts such as South Asia. Early work primarily focused on textual datasets for *Hinglish*, *Banglish*, and *Tamil–English*, targeting sentiment analysis and sequence labeling tasks (Patra et al., 2018; Singh et al., 2018; Alam et al., 2024; Nishat Raihan et al., 2023). On the speech side, datasets such as *MUCS 2021* (Diwan et al., 2021) and *Prabhupadavani* (Sandhan et al., 2022) support multilingual ASR and speech translation. More recent efforts include *MediBeng* (Ghosh, 2025) and *Switchlingua* (Xie et al., 2025). While these resources advance code-switched speech research, they generally focus on fluent speech and do not explicitly incorporate disfluency phenomena. Disfluency research has been well established in English corpora such as *Switchboard* (Godfrey and Holliman, 1997) and *FluencyBank* (Romana et al., 2024). Studies on automatic disfluency detection and correction (Amann et al., 2024) and synthetic augmentation for Indic languages (Bhat et al., 2023) highlight the importance of modeling spontaneous irregularities. However, speech-level corpora that jointly represent code-mixing and disfluency for Indic languages remain limited (Saranya et al., 2025).

Recent advances in neural TTS and large language models enable scalable synthetic corpus creation. Indic Parler TTS (Lacombe et al., 2024; Lyth and King, 2024) supports multilingual speech generation, and synthetic pipelines have been used to support low-resource ASR research (Thai et al., 2019; Yeo et al., 2026). *MixFluent* (Paul et al., 2025) has successfully demonstrated the feasibility of generating synthetic code-mixed disfluent text using LLMs, it remains strictly limited to the textual modality. MixFluent provides valuable linguistic benchmarks for Bengali-English code-mixing, but its lack of acoustic realization renders it unsuitable for training end-to-end speech systems or modeling the prosodic features of disfluency. Our dataset, *BEHE-CMDisfl*, directly extends this foundational research by bridging the gap between text and speech. We not only adapt the text generation methodology for broader coverage incorporating both Bengali-English and Hindi-English pairs, but crucially, we project these textual disfluencies into the acoustic domain using the Indic Parler TTS framework.

In ASR research, Kaldi has been widely used for low-resource and code-switched settings (Kullmann, 2016; Yılmaz et al., 2016). Although deep models perform well in high-resource scenarios (Panayotov et al., 2015; Bu et al., 2017), their effectiveness in micro-resource conditions remains lim-

ited (Dar and Pushparaj, 2026; Dhasmana et al., 2026). Indian code-mixed ASR work has largely focused on Hindi–English datasets of moderate size (Pandey et al., 2018), leaving disfluent Bengali–English speech under-explored. Taken together, existing resources either emphasize code-mixing without disfluency, disfluency without multilingual speech, or text without audio. Our dataset fill this gap that integrates Bengali–English and Hindi–English code-mixing with explicit disfluency modeling, alongside baseline ASR experiments in a micro-resource setting.

3. Dataset Development Methodology

The *BEHE-CMDisfl* dataset was constructed through a two-stage synthetic pipeline that combines (i) *ChatGPT* based generation of code-mixed disfluent text and (ii) *Indic Parler TTS* synthesis to produce the corresponding speech. Bengali and Hindi are treated as matrix languages, with English as the embedded language. The design prioritizes controlled disfluency injection, realistic code-switching behavior, and reproducibility of the generation process.

3.1. Text Corpus Design and Preparation

To ensure a generalized domain, we curated reference text prompts covering some of the major conversational contexts:

- Daily life and informal communication (e.g., greetings, personal anecdotes, job oriented discussions etc.)
- Task-oriented dialogs (e.g., customer–service interactions)
- Education and classroom discussions among teachers and students
- Healthcare and teleconsultation contexts
- Social media–style opinions, commentary and discussions about sports and entertainment

This step is a way to explore the intersection of code-mixing and disfluency in bilingual speech and text, with a focus on understanding how LLMs handle code-mixed disfluent utterances. One of the primary objectives was to explore LLMs' ability to generate code-mixed disfluent sentences and to address the lack of high-quality code-mixed disfluent corpora, particularly for Indic languages.

3.2. LLM-Driven Text Generation

To generate real and natural code-mixed disfluent text, we used a LLM, specifically, ChatGPT (**GPT-5 model**) using zero shot and few shot prompting techniques. A typical prompt used to generate a BE-CM disfluent textual conversational data is shown below:

> You are a dialog generation assistant. Generate realistic, informal conversations between multiple speakers in a South Asian context (Bengali-English), incorporating the following instructions:
>
> 1. Speaker Roles and Identity
>
> - Each line should start with the speaker's name followed by a colon, e.g., "Arjun: ..."
> - Maintain consistent personality for each speaker:
>   - Speaker A: friendly, informal, slightly humorous
>   - Speaker B: polite, thoughtful, sometimes hesitant
> - Include a variety of speakers with different age, gender, and occupation backgrounds.
>
> 2. Code-Mixing Requirements
>
> - Blend Bengali with English naturally within sentences.
> - Code-mixing should occur at the word level, phrase level, or mid-sentence.
> - Ensure frequent switching for realism, but do not overuse English.
>
> 3. Disfluency Requirements
>
> - Insert natural disfluencies like:
>   - Filled pauses: "uh", "umm", "মানে", "আচ্ছা", "you know", "hmm", "err", etc.
>   - Repetitions: repeating words or phrases
>   - Hesitations: restarting sentences, partial words
> - Include disfluencies in roughly 15
> - Disfluencies should appear randomly but contextually plausible.
>
> 4. Conversation Context
>
> - Base dialogs on everyday scenarios:
>   - Catching up with friends

- Work or office discussions
- Academic or student-related interactions
- Social plans or small talk about sports, fils, tours, etc.

- Keep conversations casual and natural.

5. Output Formatting

- Format each utterance as: SpeakerName: <utterance>
- Avoid narration or meta-comments.
- Maintain proper punctuation.
- Use Unicode Bengali characters where applicable.
- Do not translate English words unnecessarily; retain natural code-mixing.

6. Length

- Each conversation should contain 15-25 lines of dialog.

Example Instruction:
"Generate a casual conversation between two friends, Kabir and Aditi, where they catch up about work and weekend plans. Include filled pauses and spontaneous repetitions. Use Bengali-English code-mixing naturally."

We used this structured prompt in both zero-shot and few-shot settings, adding examples to stabilize speaker consistency and turn-taking. Explicit constraints on roles, code-mixing, and disfluency show a balanced structural control with natural flow. We performed no manual rewriting beyond basic formatting checks (e.g., script consistency and structural validation). All generated dialogs are available in the repository[3]. A brief portion of the structural output of our prompting strategy is shown in the below example, showcasing how the model realizes conversational disfluencies within a code-mixed Bengali–English context.

> **Arjun:** Ohh, hi... hi অদিতি, umm কেমন আছো? মানে, এখানে দেখা হবে ভাবিনি, actually…
> **Aditi:** Hey, hey অর্জুন! আমি আছি umm, আমি ভালো- I mean, আমার ভালোই যাচ্ছে এখন. তুমি- err, you?
> **Arjun:** Oh, আমি? আমি তো... মানে, আমি ঠিকই আছি. কাজের চাপটা- uh, কাজের চাপ বেশি এখন তবে... but it is okay.
> **Aditi:** Exactly… exactly, বুঝি. আমার-ও... আমার-ও actually আগের week-এ deadline ছিল- ওই যে... um, sorry, কি বলছিলাম? হ্যাঁ, deadline-টা...একদম impossible লাগছিল কিন্তু হয়ে গেলো somehow.

3.3. Speech Synthesis with Indic Parler TTS

The generated dialogs were converted into speech using Indic Parler TTS developed by AI4Bharat[4], a multilingual neural TTS framework supporting multiple Indic languages. The model was selected due to its support for Indic scripts and its ability to process mixed-script inputs, which is essential for Bengali–English and Hindi–English code-mixed text. Speaker profiles were defined through textual descriptions and supplied to the TTS model to simulate inter-speaker variability across dialog turns. This allowed consistent voice characteristics within a conversation while preserving multi-speaker structure. For instance, a profile might be:

> **Example Speaker Profile (Aditi):** "A young adult female from West Bengal with a friendly, natural voice. Speaks in a casual tone typical of a Kolkata student. Bengali words sound native and informal, while English terms are clear with a Bengali accent. Incorporates natural pauses and fillers (*um*, *hmm*, *achha*) at switching boundaries to maintain a steady, expressive conversational rhythm."

Each generated text files (as described in the previous section) was processed into speech according to Algorithm 1. Utterances were segmented into chunks of at most 25 words prior to synthesis. This chunking strategy was adopted to improve waveform stability and prevent degradation in longer inputs. The synthesized segments were then concatenated to form a single audio file per dialog. No manual post-editing was performed on the generated waveforms beyond structural validation and file consistency checks. The generated speech data are available on this repository[5].

3.4. Statistical Analysis of the Dataset

In this section we discuss about the statistical analysis of both the textual data and the corresponding speech data and introduce some of the evaluation metrics used.

3.4.1. Evaluation Metrics

Some of the evaluation metrics used are:

[3] https://github.com/anonrpd/BEHE-CMDisf/tree/main/text_data

[4] https://ai4bharat.iitm.ac.in/

[5] https://github.com/anonrpd/BEHE-CMDisf/tree/main/speech_data

Algorithm 1 Text-to-Speech Generation for Disfluent Bengali-English/Hindi-English Code-Mixed dialogs

Require: `input_folder`: directory of dialog `.txt` files; `speaker_profiles`: speaker descriptions
Ensure: `output_folder`: generated `.wav` audio files

for each file F in `input_folder` **do**
  Initialize `audio_segments` list
  Read all lines from F
  for each line in F (up to MAX_LINES) **do**
    Parse line $\rightarrow$ `speaker_name`, `utterance_text`
    if `speaker_name` not in `speaker_profiles` **then**
      skip line
    end if
    `speaker_desc` = `speaker_profiles[speaker_name]`
    Split `utterance_text` into chunks ≤ 25 words
    for each chunk **do**
      Generate audio using `model.generate(tokenized_speaker_desc, tokenized_chunk)` $\rightarrow$ `audio_waveform`
      Append `audio_waveform` to `audio_segments`
    end for
  end for
  if `audio_segments` not empty **then**
    Concatenate `audio_segments` $\rightarrow$ `final_audio`
    Save `final_audio` as `.wav` in `output_folder`
  end if
end for

1. **Mean Filler Percentage (%)**: Proportion of tokens in an utterance that are filler words (e.g., *uh*, *um*, *ah*).
2. **Mean Repetition Percentage (%)**: Percentage of tokens that are repeated immediately or within a short span.
3. **Mean Restart Percentage (%)**: Fraction of utterances exhibiting restart phenomena
4. **Mean Code-Switch Percentage (%):** Ratio of words from the embedded (non-matrix) language to total words, reflecting the degree of code-switching.
5. **Speech Ratio:** Proportion of the total duration of the audio that contains active speech segments (excluding silence).
6. **Speech Ratio Range:** Difference between the maximum and minimum speech ratio across utterances in a corpus.

3.4.2. Text Data Analysis

The corpus includes 77 Bengali–English (BE-CM) and 40 Hindi–English (HE-CM) code mixed disfluent text dialog files, covering a range of conversational contexts. As shown in Table 1, Bengali–English dialogs contain an average of $\sim$13.6 words per utterance, while Hindi–English dialogs average $\sim$11.8 words. These values indicate moderately sized conversational turns across both subsets, with comparable structural organization.

Feature	BE–CM Text Data	HE–CM Text Data
Total Files	77	40
Mean Utterances per File	31.62	44.32
Mean Words/File	392.42	520.05
Mean Words per Utterances	13.59	11.77
Mean Filler %	0.25	0.24
Mean Repetition %	0.37	0.14
Mean Restart %	1.35	1.22
Mean CS %	30.94	34.46

Table 1: Summary statistics of the Bengali–English and Hindi–English disfluent code-mixed text corpora.

Disfluency markers are present at controlled levels throughout the corpus. Filled pauses account for approximately 0.24

Figure 1 illustrates distinct code-mixing behaviors between the datasets. The Bengali-English data exhibits high variability (4.68%-62.93%), suggesting context-dependent heterogeneity, whereas the Hindi-English dataset shows a narrower, more consistent range (22.44%-55.4%). This contrast highlights the differential nature of the generated text, likely reflecting the underlying distribution of the LLM's training data.

3.4.3. Speech Data Analysis

The Bengali–English code mixed (BE-CM) speech corpus comprises 77 recordings with mean duration of 195.8 s, while the Hindi–English code mixed (HE-CM) set contains 40 recordings with mean duration of 224.4 s. Table 2 clearly shows that both subsets exhibit high speech ratios (>0.90), indicating minimal silence and stable conversational flow. Average speech rates fall within 2.26–2.34 words

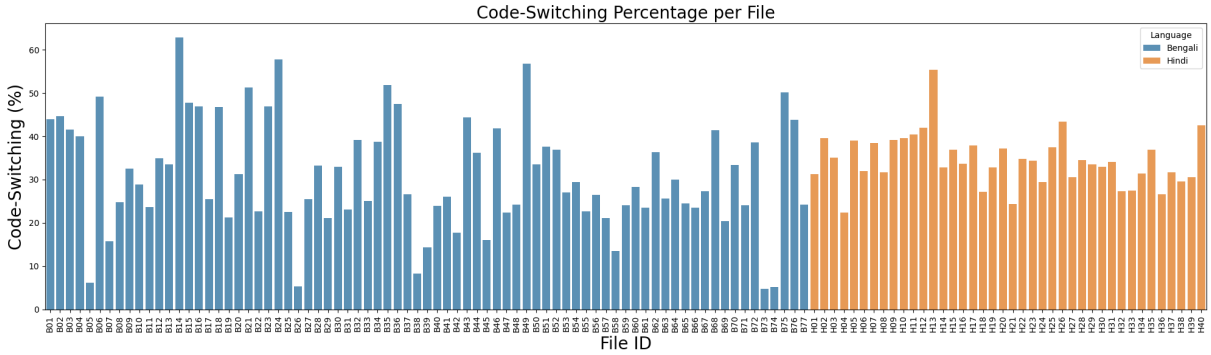

Figure 1: Code-Mixing Percentage Per Text File for Both the Languages.

per second, consistent with natural bilingual conversational tempo. The relatively low standard deviation in speech rate suggests uniform articulation speed across synthesized speakers.

Feature	BE-CM Speech Data	HE-CM Speech Data
Total recordings	77	40
Duration range (s)	108–304	136–329
Average duration (s)	195.8	224.4
Mean speech ratio	0.90	0.94
Speech ratio range	0.82–0.95	0.91–0.97
Estimated words per file	438.5 ± 110	525.4 ± 120
Speech rate (words/s)	2.26 ± 0.07	2.34 ± 0.05

Table 2: Comparative summary of Bengali-English and Hindi-English speech corpora.

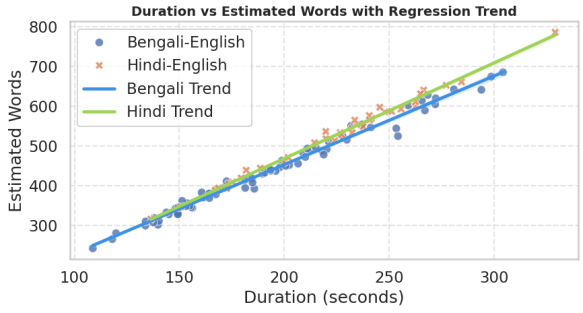

Figure 2: Relationship between speech duration and estimated word count for BE and HE datasets.

The plot shown in figure 2 shows the scatter plot of *Duration vs Estimated Words* that complements this by showing a consistent relationship between speech duration and word output for both the code-mixed disfluent datasets, with regression trends highlighting differences in speaking rate and verbosity. Together, these results indicate that our dataset maintains controlled disfluency injection at the textual level, stable acoustic realization at the speech level, and meaningful variation across the two code-mixed language pairs. This combination makes the dataset appropriate for low-resource ASR benchmarking and exploratory studies on disfluency-aware modeling.

4. Experimental Validation

To assess the efficacy of the BEHE-CMDisfl corpus in downstream tasks, we established a baseline ASR system using the Kaldi toolkit. Our objective was to determine if a model trained on this purely synthetic, disfluent data could successfully learn to decode code-mixed speech and, crucially, preserve disfluency markers rather than treating them as noise. We selected a 1.3 hour subset from the BE-CM disfluent data for this study to rigorously evaluate the pipeline's effectiveness under severe data constraints typical of cold-start scenarios.

4.1. Experimental Setup

We utilized a GMM-HMM pipeline, which remains a robust choice for micro-resource regimes where deep learning models often fail to generalize. The dataset was partitioned into a training set (12,006 words) and a held-out test set (1,236 words), maintaining an approximate 90/10 split to ensure rigorous evaluation. We employed a standard feature extraction pipeline (MFCCs + deltas + delta-deltas) and trained a progression of Monophone, Triphone, and LDA-MLLT models.

4.2. Results and Analysis

Table 3 summarizes the WER performance across different acoustic modeling stages.

Model Configuration	WER (%)
Monophone	51.14
Triphone (Tri1)	39.23
LDA-MLLT (Tri2b)	38.33
Wide-Beam Decoding (17.0)	**37.74**

Table 3: ASR performance (WER %) on the Bengali-English disfluent code-mixed subset using the Kaldi GMM-HMM baseline. The system achieves a best-case WER of 37.74% with wide-beam decoding.

The results indicate that the synthetic data possesses sufficient phonetic consistency to train a viable acoustic model from scratch. Notably, a simpler Triphone model achieved a WER of 39.23%, suggesting that for micro-resource synthetic data, lexicon consistency is often more critical than model complexity. The best performance (WER of 37.74%) was achieved by widening the decoding beam to account for the higher perplexity at code-switching points.

Achieving a WER of 37.74% is particularly significant given the micro-resource regime (1.3 hours) where deep learning benchmarks typically struggle to generalize. For context, recent studies on similar low-resource Indic languages using advanced Wav2Vec2 architectures reported WERs as high as 47.10% even with significantly more data (10 hours) (Dar and Pushparaj, 2026). Our results demonstrate that for extremely scarce, disfluent code-mixed data, a rigorously normalized GMM-HMM pipeline performs better than the theoretical baselines of deep neural models which are prone to severe overfitting in cold-start scenarios.

4.3. Disfluency Retention

A key objective was to evaluate if the model could capture synthetic disfluencies. Unlike standard systems that often filter filled pauses, our GMM-HMM baseline successfully retained these tokens. Table 4 highlights two examples where the model correctly transcribed filled pauses (*"um"*), repairs (*"i mean"*), and repetitions (*"oh oh"*). This confirms that the synthetic data provides sufficient spectral evidence to treat disfluencies as valid lexical units rather than background noise.

4.4. Impact of Normalization

We also observed that rigorous text normalization was essential. Initial experiments with raw, inconsistent transliterations (e.g., **'ar'** vs **'aar'** for the Bengali word **আর**) yielded a higher WER of 42.74%. Standardizing the lexicon improved this to 38.33%, highlighting that the quality of the synthetic text prompts is as important as the audio quality itself.

Case	Type	Transcript Sequence
1	Ref	*... **um i mean** ekta choto reunion*
	Hyp	*... **um i mean** ekta choto reunion*
	Status: Correctly identified repair & filled pause	
2	Ref	*acha **oh oh***
	Hyp	*acha **oh oh***
	Status: Correctly identified repetition	

Table 4: Successful recognition of disfluent markers in the BE-CM disfluent synthetic test set.

5. Applications and Uses

The BEHE-CMDisfl corpus is designed to address the scarcity of training data for conversational Indic speech technologies. Based on our experimental validation, we identify three primary applications:

- **Robust ASR Training in Low-Resource Regimes:** As demonstrated by our Kaldi baseline results, the corpus serves as a critical resource for bootstrapping ASR systems in micro-resource scenarios ($<$2 hours). It enables the training of acoustic models that do not merely treat disfluencies as noise but learn to transcribe them as valid lexical tokens (e.g., filled pauses, repetitions), which is essential for accurate transcriptions of spontaneous dialogue.

- **Disfluency Detection and Repair:** The corpus provides explicit, aligned examples of disfluent phenomena (restarts, hesitations) in code-mixed contexts. This structured data supports the development of supervised disfluency detection modules that can be integrated into end-to-end ASR pipelines to improve readability and downstream processing.

- **Conversational TTS Modeling:** The paired text–speech data, generated via the Indic Parler TTS, offers a controlled environment for studying prosodic modeling at code-switching points. Researchers can use this dataset to analyze and improve the naturalness of synthetic voices when navigating the complex prosody of bilingual sentence structures.

6. Conclusion and Future Work

We presented BEHE-CMDisfl, a synthetic, reproducible Bengali-English and Hindi-English code-mixed speech corpus with explicit annotations for disfluency phenomena. Our experimental validation using a Kaldi GMM-HMM baseline demonstrates that the dataset can successfully train disfluency-aware acoustic models, achieving a WER of 37.74% and correctly identifying filled pauses and repetitions for the BE-CM disfluent data. This confirms that synthetic data, when generated with consistent lexicons, is a viable stopgap for low-resource Indic speech research. While our synthetic pipeline effectively captures disfluency markers for ASR training, we view it as a targeted computational framework for modeling specific linguistic phenomena. We acknowledge that synthetic audio serves to approximate conversational patterns and may not fully replicate the complex acoustic variety of natural human speech. Future iterations of this study will include statistical significance testing and cross-validation across multiple training seeds to further verify the stability of our WER findings.

Our immediate road map focuses on validating and extending this resource:

- **Real-Speech Validation:** To evaluate the generalizability of our findings, we plan to curate a 'gold-standard' dataset consisting of 3-5 hours of natural conversational speech. This benchmark will allow us to formally quantify the 'Sim-to-Real' gap and assess how effectively our synthetic-trained models generalize to noisy, real-world environments.
- **Speech-to-Speech Translation:** We plan to explore using this disfluent data to train translation models that can normalize disfluent code-mixed speech into fluent English speech.
- **Broader Language Coverage:** The pipeline will be extended to other morphologically rich Indic languages, such as Tamil, Marathi, and Telugu, to test cross-lingual generalization. Also we will train and test on the Hindi-English disfluent code-mixed dataset as well.

7. Summary

7.1. Summary of Contributions

We present **BEHE-CMDisfl**, a synthetic, reproducible Bengali–English and Hindi–English code-mixed speech corpus with explicit annotations for disfluency phenomena and language segments. The corpus fills an important gap for multilingual, disfluent speech modeling in Indic contexts and supports ASR fine-tuning, disfluency detection, TTS research, and sociolinguistic studies.

7.2. Final Remarks

We believe **BEHE-CMDisfl** provides a practical, reproducible resource for the community while emphasizing transparency and ethical safeguards. Researchers using the corpus should validate findings on real conversational speech before deployment in safety-critical settings.

8. Ethical Considerations and Limitations

We recognize both the benefits and potential risks of generating synthetic, code-mixed, disfluent speech. Key ethical aspects and mitigations are summarized below:

Transparency: All generation steps—including LLM prompts, TTS model versions, and post-processing scripts—should be fully documented to ensure reproducibility and accountability.

Misuse Risks: Synthetic speech can be exploited for voice spoofing or misinformation. To mitigate such misuse, dataset releases must include clear usage guidelines, watermarking procedures, and, where possible, benchmarks for deepfake detection.

Bias and Representativeness: Both text generation and TTS systems inherit demographic and linguistic biases. We report speaker coverage, avoid mimicking real individuals, and caution against overgeneralization across underrepresented dialects.

Evaluation Limits: Synthetic disfluencies may not fully capture natural conversational patterns; results obtained from synthetic data should be validated on real-world speech to ensure generalization.

Privacy and Consent: As no real voices are used, privacy risks are minimal. However, the potential for misuse (e.g., impersonation) warrants controlled access and restrictive licensing.

References

Jordi Adell, Antonio Bonafonte, and David Escudero. 2006. Disfluent speech analysis and synthesis: A preliminary approach. In *Proc. 3rd Int. Conf. on Speech Prosody*, pages 1–4.

Sadia Alam, Md Farhan Ishmam, Navid Hasin Alvee, Md Shahnewaz Siddique, Md Azam Hossain, and Abu Raihan Mostofa Kamal. 2024. Bnsentmix: A diverse bengali-english code-mixed dataset for sentiment analysis. *arXiv preprint arXiv:2408.08964*.

Robin Amann, Zhaolin Li, Barbara Bruno, and Jan Niehues. 2024. Augmenting automatic speech recognition models with disfluency detection. pages 224–231.

Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity*. Routledge.

L. Besacier, E. Barnard, A. Karpov, and T. Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

Vineet Bhat, Preethi Jyothi, and Pushpak Bhattacharyya. 2023. Adversarial training for low-resource disfluency correction. *arXiv preprint arXiv:2306.06384*.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *Proc. O-COCOSDA*, pages 1–5.

M. A. Dar and J. Pushparaj. 2026. A wav2vec2 model-based automatic speech recognition system for low-resource kashmiri language. *International Journal of Speech Technology*, 29(1):2.

A. Dhasmana, A. Srivastava, and D. Chiang. 2026. Dialect matters: Cross-lingual asr transfer for low-resource indic language varieties. *arXiv preprint arXiv:2601.04373*.

Anuj Diwan, Rakesh Vaideeswaran, Sanket Shah, Ankita Singh, Srinivasa Raghavan, Shreya Khare, Vinit Unni, Saurabh Vyas, Akash Rajpuria, Chiranjeevi Yarra, et al. 2021. Multilingual and code-switching asr challenges for low resource indian languages. *arXiv preprint arXiv:2104.00235*.

Promila Ghosh. 2025. Medibeng (revision b05b594).

J Godfrey and E Holliman. 1997. Switchboard-1 release 2: Linguistic data consortium. *Switchboard: a user's manual*.

A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

T. D. Harya. 2018. Sociolinguistics: Code switching and code mixing. *Lentera: Jurnal Ilmiah Kependidikan*, 11(1):87–98.

Eunhee Kim. 2006. Reasons and motivations for code-mixing and code-switching. *Issues in EFL*, 4(1):43–61.

E. Kullmann. 2016. Speech-to-text for swedish using kaldi. Master's thesis, KTH Royal Institute of Technology.

Rohit Kundu, Preethi Jyothi, and Pushpak Bhattacharyya. 2022. Zero-shot disfluency detection for indian languages. In *Proceedings of the 29th international conference on computational linguistics*, pages 4442–4454.

Y. Lacombe, V. Srivastav, and S. Gandhi. 2024. Parler-tts. GitHub repository.

D. Lyth and S. King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *arXiv preprint arXiv:2402.01912*.

M. Mohri, F. Pereira, and M. Riley. 2008. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer.

Melissa G Moyer. 2002. Bilingual speech: A typology of code-mixing.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. Sentmix-3l: A bangla-english-hindi code-mixed dataset for sentiment analysis. *arXiv e-prints*, pages arXiv–2310.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 5206–5210.

A. Pandey, B. M. L. Srivastava, and S. Sitaram. 2018. Adapting monolingual resources for code-mixed hindi-english speech recognition. In *Proc. IEEE ICASSP*.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017. *arXiv preprint arXiv:1803.06745*.

Aryan Paul, Tapabrata Mondal, Dipankar Das, and Sivaji Bandyopadhyay. 2025. Generating and analyzing disfluency in a code-mixed setting. *Recent Advances in Natural Language Processing (RANLP)*, pages 915–924.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, et al. 2011. The kaldi speech recognition toolkit. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Amrit Romana, Minxue Niu, Matthew Perez, and Emily Mower Provost. 2024. Fluencybank timestamped: An updated data set for disfluency detection and automatic intended speech recognition. *Journal of Speech, Language, and Hearing Research*, 67(11):4203–4215.

Jivnesh Sandhan, Ayush Daksh, Om Adideva Paranjay, Laxmidhar Behera, and Pawan Goyal. 2022. Prabhupadavani: A code-mixed speech translation data for 25 languages. *arXiv preprint arXiv:2201.11391*.

S Saranya, B Bharathi, S Gomathy Dhanya, and Aishwarya Krishnakumar. 2025. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

Elizabeth Shriberg. 2001. To 'errrr'is human: ecology and acoustics of speech disfluencies. *Journal of the international phonetic association*, 31(1):153–169.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the seventh named entities workshop*, pages 27–35.

S. Sitaram, K. R. Chandu, S. K. Rallabandi, and A. W. Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Bao Thai, Robert Jimerson, Dominic Arcoraci, Emily Prud'hommeaux, and Raymond Ptucha. 2019. Synthetic data augmentation for improving low-resource asr. In *2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, pages 1–9.

S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, et al. 2018. Espnet: End-to-end speech processing toolkit. In *Proc. Interspeech*.

Peng Xie, Xingyuan Liu, Tsz Wai Chan, Yequan Bie, Yangqiu Song, Yang Wang, Hao Chen, and Kani Chen. 2025. Switchlingua: The first large-scale multilingual and multi-ethnic code-switching dataset. *arXiv preprint arXiv:2506.00087*.

Y. H. Yeo, Y. Hu, S. Gopal, Y. Peng, H. Liu, and E. S. Chng. 2026. Improving code-switching speech recognition with tts data augmentation. *arXiv preprint arXiv:2601.00935*.

Emre Yılmaz, M. Andringa, S. Kingma, J. Dijkstra, F. van der Kuip, H. van de Velde, et al. 2016. Longitudinal speaker clustering and identification for frisian-dutch code-switching. In *Proc. Interspeech*, pages 3668–3672.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional lstm. *arXiv preprint arXiv:1604.03209*.